Most recent update: July 3, 2000

Recommended Practices for AP 210

Electronic Assembly, Interconnect, and Packaging Design

[Coordinated with pre-publication IS]

****DRAFT****

Revision page

Updated 1.5.1.9--1.5.12 (categorizing products, etc.) for clarity. no changes in formal list elements

Updated 1.5.4 (properties to products) to specifically identify the methods to assign data element type definitions and data element type values to categories and to specific products.

Updated 1.5.20 for 2D -- 3D conversion in and out of AP 210 context.

5/30/99 editorial updates in other areas, rework 1.5.20

6/5/99 removed references to package alternates for mirroring as required. Now the recommendations assume CAD system can mirror.

6/16/99 re-inserted figures 1 and 2.

2/23/2000 revised, restarted change bars;

3/14/2000 reorganized, incorporated 2d 3d integration
3/15/2000 added links to pdf and ppt files.
4/18/2000 updated 2d 3d integration, added examples
4/24/2000 added missing figure 35, re-oriented figure 22
4/28/2000 updated list of allowed standard categories clause 2.3.1.10
5/19/2000 updated graphics for 2d/3d material
5/21/2000 added OpenGl pdf file as reference
6/10/2000 corrected printed circuit assembly physical design view
6/19/2000 added figures showing 3D placementof orientation symbology
7/3/2000 corrected figures Interconnect Module Edge and Interconnect Module Edge Geometry and Assembly Component Location

1 Introduction and Overview of STEP[1]

1.1 About AP 210

STEP Application Protocol (AP) 210 (Electronic assembly, interconnect, and packaging design) provides the data structures for the exchange of electromechanical design, requirements, functional specification, and configuration identification with configuration control data either with or without associated 2D or 3D model information. This international standard was developed under the auspices of the International Organization for Standardization (ISO) and is one of a series of parts comprising the full Standard for the Exchange of Product model data (STEP) standard known as ISO 10303.

AP 210 was developed to represent the domain which is described in its scope section, and is centered on the requirements definition/design/analysis/manufacturing phase of electromechanical products. Other APs (currently under development or proposed) will carry the data in AP 210 forward through the product life cycle.

1.2 Using This Document

The purpose of this document is to provide a usage guide for industry and to ensure the consistency of mappings to the AP 210 schema from existing software application data structures, other Application Protocols, and related industry standards. This document assumes that the reader has at least a rudimentary knowledge of both AP 210 and its associated application domain (for basic conceptual information on AP 210, see the AP 210 Concept of Operations). The figures in this document are intended to provide a navigational view of portions of the AP with boxes representing entities, lines being relationships, and arrow heads (or bubbles) indicating the direction of dependency. An entity "points to" other entities upon which it is dependent. The arrow head or bubble is on the end adjacent to the supporting entity. See ISO 10303-11 annex D for more information.

This document will provide pre- and post-processor recommendations where attributes from the conceptual STEP data models may not actually have values in the AP 210 application domain, as documented in AP 210. The terms pre-processor and post-processor refer to the applications which write and read the AP 210 data respectively. In these recommendations, the term "no standard mapping" means there is no mapping defined in the AP's ARM to AIM mapping table for the data. The term "no standard string" means that either:

- there is no documented consensus on string values defined in the AP's ARM to AIM mapping table,
- there is no rule in the AP's AIM to constrain the allowed values.

This document is not intended to replace the AP ARM to AIM mapping table specified in clause 5 of AP 210 or the requirements specified in clause 4 of AP 210 or the rules specified in clause 5 of AP 210.

1. filename: c:\Projects\recommended_practices\AP 210_recpracfeb.doc

1.3 General STEP Application Guidelines

There are some constructs which have global applicability across all data in a STEP exchange. These constructs relate to the file header for physical file exchanges, data definitions within the file related to the AP itself, and fundamental constructs which contain the information for people, organizations, dates, times, approvals, security classifications, and units of measure. References to ISO 10303 documents herein are to the latest available unless otherwise specified, and include any technical corrigendum announced. Refer to the following URL for currently available editions: SOLIS ....

1.3.1 The STEP Physical File

One way to externalize AP 210 data is via a physical file which is an ASCII encoding of the data based on the EXPRESS constructs in AP 210 as mapped through ISO 10303-21. This document will not deal with this in any detail. The user is referred to ISO parts 10303-11 for definitions of EXPRESS constructs and 10303-21 for information on how to map the EXPRESS constructs in AP 210 to the physical file. (See ******* for implementors agreements that affect physical files.) Public domain toolkits and commercial software are available to automate the task. See the following URLs....NIST/ EPM/STEP Tools/ProStep. This document does not form an endorsement of any vendor offering.

Post-processor Recommendations: When reading a physical file, post-processors should note all errors found during the reading of the file. It is recommended that post-processors provide options to the user on whether to continue when an error is encountered. There are no recommendations on what a post-processor should do with erroneous data. This is left to the discretion of the implementor. If the implementor elects to correct erroneous data, the post-processor should inform the user (as above) of the erroneous data and what correction was made.

1.3.2 The STEP SDAI

The other way to externalize AP 210 data is via an SDAI interface which is a procedural interface based on the EXPRESS constructs in AP 210 as mapped through ISO 10303-22 and one of ISO 10303-23 through ISO 10303-26. "C", "C++", Java, IDL bindings are currently available. The document will not deal with this in any detail. The reader is referred to the above documents for information on how to map the EXPRESS constructs in AP 210 into a specific SDAI implementation. Public domain and vendor supplied toolkits are available to automate this task. The previously specified URLs also provide links to support SDAI implementations.

SDAI Read Recommendation: When extracting data from a database, processors should note all errors found. It is recommended that processors provide options to the users on whether to continue when an error is encountered. There are no recommendations on what a processor should do with erroneous data. This is left to the discretion of the implementor. If the processor elects to correct erroneous data, the processor should inform the user (as above) of the bad data and what correction was made.

1.3.3 The W3C XML

Currently various research activities are underway to determine the optimal way to externalize AP 210 data over the web using XML. This document will be updated as a consensus develops.

1.3.4 AP Identification and Contexts

STEP is an intelligent data standard and as such the representation of the data for an AP identifies the AP data structure through computer sensible data. This is done through the entities **application_context** and **application_protocol_definition**. These and related entities are described pictorially in **Figure 1**.

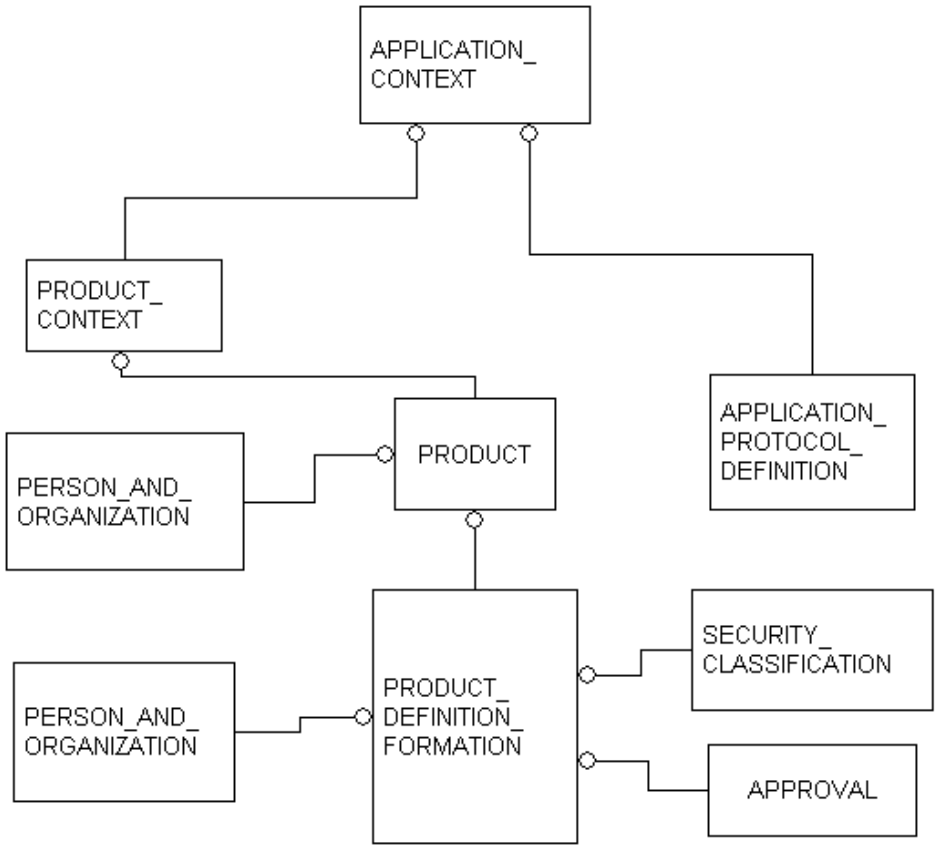


FIGURE 1 AP identification and product context

The **application_context** entity identifies the application which defined the data. The **application** attribute, based on its definition in ISO 10303-41, should have the value "electronic assembly interconnect and packaging design" as this is the application domain AP 210 is meant to cover.

Note: industry usage may indicate this is equivalent to the broader electro-mechanical design or electronic packaging design.

The **application_protocol_definition** entity further identifies the AP. For AP 210, the **status** attribute, based on its definition in ISO 10303-41, should have the value "draft international standard". The **application_interpreted_model_schema_name** attribute should have the value "electronic_assembly_interconnect_and_packaging_design" based on an AP 210 restriction. The

application_protocol_year attribute should have the value "1998", based on its definition in ISO 10303-41.

The application identified by the **application_context** entity is broken down into elements in STEP. In AP 210, these elements are contexts. For AP 210, the valid context entities are **product_context**, **product_definition_context**, and **product_concept_context.** Figure 1 illustrates **product_context**. Figure 2 illustrates **product_definition_context.** For clarity, some of the entities required to establish the relationships from the following entities are not included: **person_and_organization**, **security_classification**, **approval**.

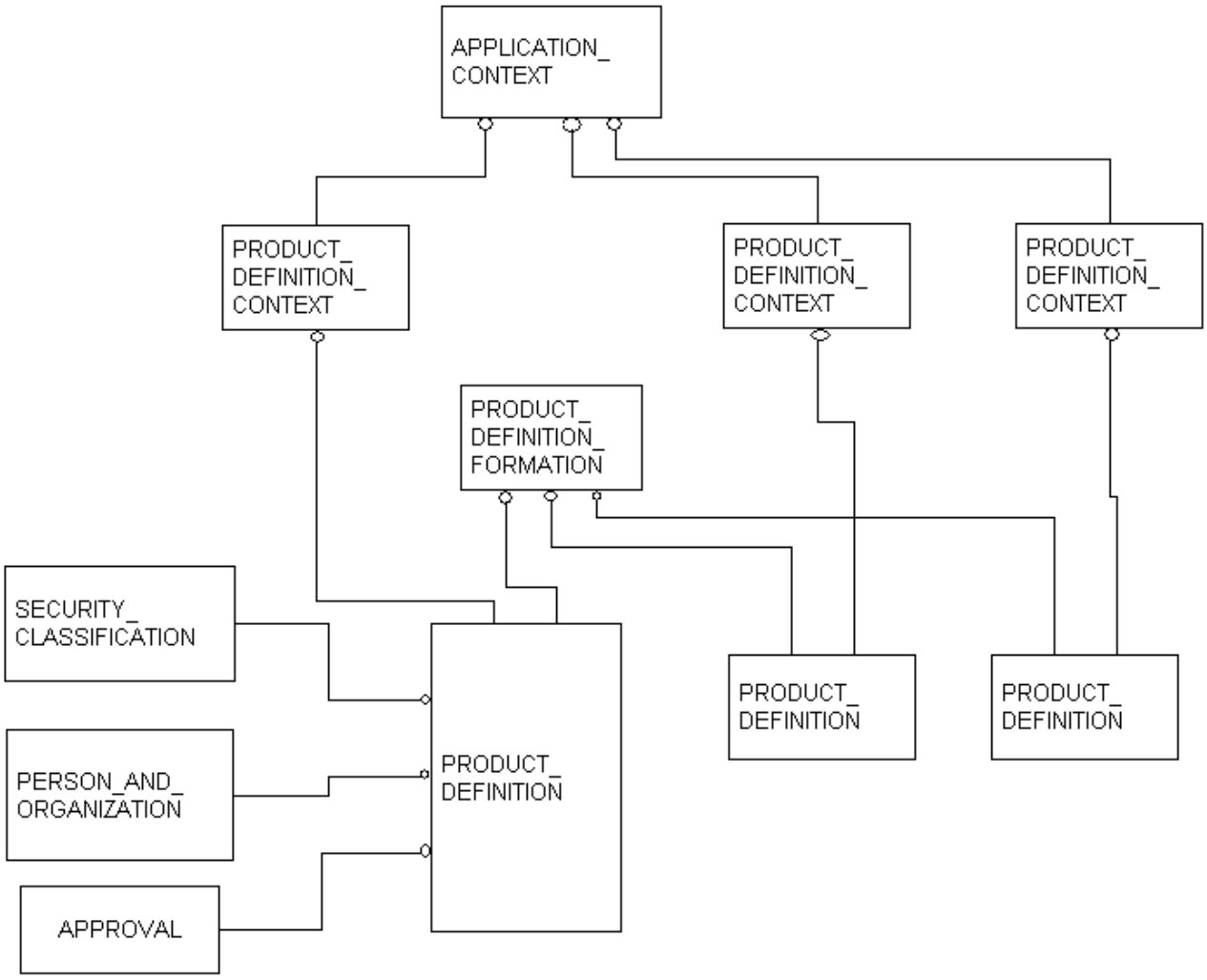


FIGURE 2 Product definition context

2 AP 210 Specific Application Guidelines

2.1 AP 210 Specific Contexts

2.1.1 Product_context

The **product_context** entity identifies a high level engineering discipline as a context. Often this discipline is considered culturally or from a business perspective to be the discipline responsible for the product. This context data has no explicit or implied relation to the "views" of the product provided by **product_definition** enitities. For AP 210 which supports multiple disciplines for one product, there is no required subtype of product_context. There is no standard character string set for the value for the attribute **discipline_type**. If populated, values such as:

- "analog",
- "device",
- "digital",
- "electrical",
- "electrical-mechanical",
- "industrial",
- "logistics",
- "mechanical",
- "microwave",
- "packaging"
- "photonic".
- "rf",
- "system",
- "video",

are recommended. If used, it is recommended that the **discipline_type** attribute indicate the primary discipline type of the organization responsible for the product. Products in the domain of AP 210 may be physical or may be intellectual (a set of requirements for example). An identifier for this discipline will likely appear in the product category network and there may be a product_definition associated with this discipline. AP 210 permits industrial engineering organizations to consider their data definitions as 'design' data. A typical example of this is the 'panel' designed by an industrial engineering organization in a manufacturing company. This is just as valid a use of AP 210 as is a single circuit card design. The product categories, product identifiers, and organization identifiers provided by AP 210 allow the context to be preserved. Note that terms such as "MCM-D, MCM-L" are considered in AP 210 to be product categories and not discipline categories.

Pre-processor Recommendations: Implementation agreements may specify values for the **discipline_type** attribute.

Post-processor Recommendations: Post-processors should preserve string values.

2.1.2 Product_definition_context

The **product_definition_context** entity identifies the context for product definitions in the AP 210 domain. The **product_definition_context** entities will establish the viewing perspective and therefore the requirements source for **product_definition** entities. Each **product_definition** entity may have its own instance of **product_definition_context** in order to unambiguously define the context. There is no standard set of character strings for the **life_cycle_stage** attribute of **product_definition_context**. The **life_cycle_stage** attribute is mapped to the AP 210 domain requirement Ee_product_definition.discipline_id. It is recommended that the engineering discipline that created the product_definition_context (and the associated **product_definition**) be identified with this attribute. The **life_cycle_stage** attribute and the **approval level** attribute are the only attributes available to distinguish **product_definition** discipline and life cycle concepts. Use them wisely. The recommended values for this attribute are:

- "electrical",
- "industrial",
- "logistics",
- "maintainability",
- "mechanical",
- "reliability",
- "software",
- "system",

followed optionally by either "definition, analysis", followed optionally by one of:

- "as required",
- "as designed",
- "as planned",
- "as built",
- "as maintained".

The intent is that there would be one or more strings that identify the discipline and purpose of the context. It will be necessary to include these strings in exchange agreements since they are not part of the standard. In some cases, there are non-sensical and redundant combinatations of the values entirely possible. In some cases where shape data is present (i.e., for subtypes of **physical_unit** or subtypes of **part_template_definition**), additional standard contextual information is provided. The additional data (included as several **descriptive_representation_item**) describes:

- material condition,
- purpose,
- usage environment.

The **product_definition_context** entity inherits a **name** attribute from the supertype **application_context_element**. Standard string values for the **name** attribute are:

- "design requirement",
- "functional definition",
- "functional design usage",
- "functional network design",
- "functional occurrence",
- "layout occurrence",
- "physical design",
- "physical design usage",
- "physical occurrence",
- "requirement",
- "template definition".

These values are mapped from the requirements specified in clause 4.2 of AP 210 for subtypes of Ee_product_definition. For formal definitions, the definitions in 4.2 for the application objects associated with these strings shall be referenced. Do not assume that these values map automatically onto the recommended values for the **life_cycle_stage** attribute. There is no replacement for common sense. Some examples of these are:"physical design" indicates that the context is a "design" context with a model that describes physical things. It does not specify that the associated **product_definition** has a shape associated or that the shape is a solid model (if there is a shape). "physical design usage" indicates that the context is a "usage" context; in this context there is sufficient information to incorporate the design into another design, but there may be insufficient information to produce a product. "physical occurrence" indicates that the associated **product_definition** will contain additional information specific to the occurrence of an item beyond that in the original "design" or "usage" contexts. "functional definition" indicates that the context is an abstraction or representation of some behaviour. As an example of a perhaps unique usage, a mechanical analysis application may choose to use the "functional network design" to model a physical item (certainly only valid under some simplifying assumptions).

Pre-processor Recommendations: See above.

Post-processor Recommendations: Post-processors shall interpret string values in accordance with the AP 210 mapping table.

2.1.3 Product_concept_context

The **product_concept_context** entity identifies what market segment or customers provided requirements for the data. This entity will establish the source of the requirements for **product_concept** entities.

Pre-processor Recommendations: Implementors agreements may establish values for the **market_segment_type** attribute.

2.2 Generic Product Semantics

2.2.1 People and Organizations

AP 210 represents people and organizations as they perform functions related to other data and data relationships. A **person** or a **person** in an **organization** is associated to some data or data relationship in some role indicating the function being performed.

Both people and organizations may have addresses associated with them. This is entirely optional in AP 210 and is done through the **address** entity being related to the **person** (through **personal_address**) or **organization** (through **organizational_address**).

2.2.1.1 People

AP 210 specifies information about people through the **person** entity. A **person** is identified by an **id** with other data representing their name and optionally titles which may apply to them. In populating the data, the **id** must be unique if the **person** is associated with an organization through the **person_and_organization** entity.

Pre-processor Recommendations: All pre-processors should provide values for at least the **last_name** and **first_name** attributes for the **person** entity.

2.2.1.2 Organizations

AP 210 represents groups of people (e.g. design departments, industrial engineering departments, reliability engineering departments, companies, countries, etc.) through the **organization** entity. The identification or **id** data is optional. This information can be highly important in providing unique identification to the organization or company. It is recommended that this field always be populated with unique data. The **name** attribute must contain a short identifier or acronym for the **organization**. The **description** attribute may contain the full name of the organization or a textual explanation its reason for existence.

Pre-processor Recommendations: All pre-processors should provide a unique **organization id** to eliminate ambiguities where organizations may have the same names. If the intended domain for the data is large, the reader is referred to ISO/IEC 8824-1 which can provide some guidance on creating unique identifiers. For example, if the organization typically used a CAGE identifier of "93699" and the identifiers associated with CAGE were registered by the US government (i.e., "CAGE" is defined in the US "federal stock number system"), the actual value of the **organization id** would be "COUNTRY,USA,CAGE,93699". Since the CAGE code is widely used, including traceability of "CAGE" to "federal stock number system" and thence to USA is not recommended.

Post-processor Recommendations: All post-processors should make use of any provided information in the **id** attribute to eliminate ambiguities where **organization**s may have the same name.

2.2.1.3 Roles of People and Organizations

The connection of people to organizations is accomplished through the **person_and_organization** entity. It is used to identify approvers for different aspects of the product data. It is also related to certain constructs to identify the people and organizations responsible for them and how they are responsible. This is done through the **applied_person_and_organization_assignment** entity which relates a person_and_organization or organization in some role to an entity. The role is established in the **person_and_organization_role** entity **name** attribute. Standard character strings are established for this attribute in AP 210 mapping tables and are used in the rules in clause 5 in AP 210.

Pre-processor Recommendations: Standard character strings shall be used if provided.

Post-processor Recommendations: All post-processors should make use of any provided information in the **person_and_organization_role** entity **name** attribute, when the data in the mapping table does not apply.

2.2.2 Dates and Times

AP 210 represents dates and times to record when something occurred. AP 210 requires both a date and a time for all events.

2.2.2.1 Dates

AP 210 provides one way to represent a **date**, **calendar_date**.

2.2.2.2 Time

AP 210 represents time through the entity **local_time**. As mentioned earlier, the requirement that time be provided for every date may involve the invention or defaulting of data.

The **local_time** entity references a time zone identification through the **zone** attribute. The referred to **coordinated_universal_time_offset** entity identifies the delta from the current time zone to coordinated universal time. For AP 210's application domain, this should be considered the delta in hours and minutes between Greenwich Mean Time (GMT) and the local time zone.

> *NOTE - Coordinated Universal Time is NOT exactly Greenwich Mean Time (GMT). The hour and minute offset is the same, but the second offset varies due to seasonal variations in the earth's axis orientation. The difference between GMT and coordinated universal time is on the order of .05 seconds which has essentially no effect in a configuration management (AP 210) application.*

Pre-processor Recommendations: All pre-processors should use noon in the originating time zone as a default for **local_time** when this data is unavailable. All pre-processors should view Greenwich Mean Time and coordinated universal time as equal.

2.2.2.3 Roles of Dates and Times

The connection of dates to times is accomplished through the **date_and_time** entity. It is used to identify when an approval occurred for different aspects of the product data. It is also related to certain constructs to identify the date and time something started, stopped or occurred and what started, stopped or occurred. This is done through the **applied_date_and_time_assignment** entity which relates a date and time in some role to some construct. The role is established in the **date_time_role** entity **name** attribute. The data allowed in this attribute is constrained by the **applied_date_and_time_assignment** entity or by the **applied_date_assignment** entity in the data model.

Pre-processor Recommendations: See above.

Post-processor Recommendations: Post-processors shall interpret string values in accordance with the AP 210 mapping table.

2.2.3 Approvals

2.2.3.1 Approval

There are many constructs in AP 210 which require approvals. Approving in AP 210 is accomplished by establishing an **approval** entity and relating it to some construct through a **applied_approval_assignment** entity. There are rules related to the use of the **approval** entity which require it to have an associated **approval_person_organization** and **approval_date_time**. This is sensible as an approval is normally given by someone at a certain time.

Every construct which requires an **approval** is allowed only one **approval**. This might lead to the misconception that only one person on one date/ time can approve something. This is not the case. The approval constructs in AP 210 actually designate that an approval cycle is required. This cycle may only need one signature.

The **approval level** attribute in AP 210 is mapped to the domain requirement for identification of activities within a broader life cycle stage through the AP 210 application object Ee_product_version and associated attribute life_cycle_status. The standard string values are:

- "conceptual design",
- "preliminary design",
- "detailed design",
- "final design".

"final design" is intended to allow interoperability with APs that are focused at the broad life cycle stages as it is typically assigned at design release. There are no rules in AP 210 to require these values. Exchange agreements will need to enforce these strings.

Pre-processor Recommendations: It is recommended that this attribute contain one of the strings identified above as minimal content. If no appropriate data for the **approval_role** attribute (why this person_and_organization or organization is approving) is available it is recommended that this attribute contain the value "approver". It is recommended that all **approval_person_organization** instances have associated **applied_date_and_time_assignment** entities to provide complete clarity.

Post-processor Recommendations: Industry domain agreements should enforce the allowed values for **approval level** attribute value.

2.2.3.2 Approval_status

The **approval_status name** attribute in AP 210 has a restriction on its possible values. The values shall only be "approved", "not_yet_approved", "disapproved" or "withdrawn". This restriction is enforced by the **restrict_approval_status** rule.

2.2.3.3 Approval_date_time

The **approval_date_time** records the date/ time the status was changed. It does not record (necessarily) when the approval was given by the **approval_person_organization** as there can be multiple **approval_person_organization**s related to an **approval** entity. If there is only one **approval_person_organization** and the **approval_status** is "approved", the **approval_date_time** indicates that this person/organization approved it on this date/time. When an approval event is a cycle which requires multiple people to concur on possibly differing dates/times, the dates/times are recorded through the relation of a **applied_date_and_time_assignment** entity with the **date_time_role** being "sign off date". This relation is not required in the AP. In the cycle case, the **approval_date_time** only indicates when the status of the **approval** was last changed.

2.2.4 Security

AP 210 requires that certain constructs indicate their sensitivity to the owning organization. This is accomplished by establishing the **security_classification** entity and relating it to the construct via the **applied_security_classification** entity. The classification is given in **security_classification_level name** attribute. Suggested values of this attribute are: "unclassified", "classified", "proprietary", "confidential", "secret", and "top_secret". It should be noted that the value of "classified" only indicates that the data is not unclassified. This value is used when an organization has a security classification which does not exactly match any of the other values.

A **security_classification** in AP 210 requires an approval, a person_and_organization or organization in the role of **classification_officer** and a date and time in the role of **classification_date**. It should be noted that the AP provides for indication of an expiration date for the classification by relating a date/ time in the role of **declassification_date**, but this is not required.

Pre-processor Recommendations: There is no standard mapping for the **security_classification purpose** attribute. Since there is no standard mapping in the AP 210 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate/ mutually agreed upon string. There is no standard mapping for the **security_classification name** attribute. Since there is no standard mapping in the AP 210 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate/ mutually agreed upon string.

If the **security_classification_level name** attribute is the value "classified", it is recommended that the organization's classification designation be placed in the **security_classification name** attribute. For example, if an organization had a security classification of "secret restricted", the **security_classification_level name** attribute value would have the value "classified", and the **security_classification name** attribute would have the value "secret restricted".

Post-processor Recommendations: There is no standard mapping for the **security_classification purpose** attribute. Since there is no standard mapping in the AP 210 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate/ mutually agreed upon string. If the **security_classification_level name** attribute is the value "classified", it is recommended that post-processors regard the **security_classification name** data as the identification of a special or non-standard classification. If the **security_classification_level name** attribute has a value of other than "classified", it is recommended that post-processors not assign any processing significance to the **name** attributevalue.

2.2.5 Units of Measure

AP 210 provides for a number of units of measure which can be used for quantities or determining the dimensionality of a set of information. All units of measure information supported in ISO 10303-41 are supported in AP 210.

2.2.6 Shape Representation

AP 210 provides multiple types of **shape_representation** which are grouped into conformance classes as shown in clause 6 of the AP. These classes are:

- geometrically bounded shape models:
  - geometrically bounded surface model
  - geometrically bounded 2d wireframe model
  - Curve 2d
  - Basic curve 2d
- wireframe with topology shape models which are represented by
  - **edge_based_wireframe_shape_representation**
  - **shell_based_wireframe_shape_representation**,
  - **edge_based_2d_wireframe_shape_representation**
  - **shell_based_2d_wireframe_shape_representation**,
- open shell model which is represented by **manifold_surface_shape_representation** (with additional constraints),
- constructive solid geometric model which is represented by **csg_shape_representation**
- and boundary representation model which is represented by **advanced_brep_shape_representation**.

This document will not go into detail on **shape_representation**. It will only present clarifications and practices for the different types of shape as appropriate. There are a number of potential implementor agreements in regard to shape.

2.2.6.1 Units for Shape Representation

Units are defined for a type of **shape_representation** through the use of a complex instance of **global_unit_assigned_context** and **geometric_representation_context**. When global units are used, units must be defined for **length_unit**, **plane_angle_unit**, and **solid_angle_unit**. The base units for STEP are Standard International (SI) units which are represented through the **named_unit** sub-type **si_unit**. All other units (such as English units) are represented as **conversion_based_unit** entities which reference **si_units**. Physical file examples for SI and English units can be found in appendix D on page 68. (See G.1.9 on page 83 for an implementors agreement that affects units.)

=========reference needs updating

In addition to the global measurement units described in the prior paragraph, AP 210 provides for the definition of a global gap tolerance for a shape model through the addition of **global_uncertainty_assigned_context** to the complex **representation_context** instance. This entity defines a set of **uncertainty_measure_with_unit** entities to represent various gap type measurements. Physical file examples for SI and English units of uncertainty can be found in appendix D on page 68.

=========reference needs updating

As a clarification to 10303-42, units on parametric representations are taken from the **global_unit_assigned_context** entity. They are not always degrees as might be extrapolated from reading the text of 10303-42. This is a consideration on choosing the global units for plane angles as radian units are irrational and potentially unstable.

Pre-processor Recommendations: If a pre-processor uses **global_uncertainty_assigned_context**, it should point to one **uncertainty_measure_with_unit** which should identify a **length_measure**. The value of the **name** attribute shall be "closure". The **length_measure** shall contain the value of the largest gap anticipated between elements that should be deemed coincident.

Pre-processors should use degree as the unit for **plane_angle_unit** as it is more stable than using a radian unit.

Post-processor Recommendations: Post-processors shall use the **uncertainty_measure_with_unit** value for error checking of the file where an error is a gap in the shape which is larger than the **length_measure** value.

2.2.6.2 Boundary Representation Models

This sub-section will not provide detailed information on boundary representation models. It should be noted that most AP 210 conformance class 17 implementations are providing surface, seam and intersection curves in the data. This has been noted and is being considered for future editions. This practice may also be formalized through an implementors agreement in the interim. Organizations implementing AP 210 should provide provisions for this data in their post-processors even though AP 210 has a rule which precludes it.

Pre-processor Recommendations: Pre-processors should not use **face_outer_bound** designations on closed periodic surfaces (cylinder, sphere, torus) as this designation is ambiguous. Pre-processors

Post-processor Recommendations: Post-processors should ignore the **face_outer_bound** designations on closed periodic surfaces (cylinder, sphere, torus) as this designation is ambiguous.

2.3 Product Structure and Identification

In order to define a product in AP 210, several AIM entities are used. The **product** entity establishes its identification (or part number), name (or nomenclature), and description. The **product_category** entity establishes the product in a classification family. The **product_definition_formation** entity identifies its version (or change level). The **product_definition_context** entity, in conjunction with the **product_definition** or **product_definition_with_associated_documents** entity identifies the engineering discipline view of a specific model of the product. Using those entities (and EXPRESS sub-types), the product is identified, revision controlled, and its context views are defined. **Figure 2** describes pictorially the relationships among the entities needed to define a product in AP 210 at a high level. These entities and relationships are necessary in order for AP 210 to support various configuration control methodologies. The **product_related_product_category** is used to differentiate between assembly module and interconnect module for example in the "physical design" context.

2.3.1 Identifying Products

2.3.1.1 The Product Entity

AP 210 deals with all parts as **product**s. The part number for a part is stored in the **id** attribute. The nomenclature or name of the part is stored in the **name** attribute. If there is an expanded name or description of the part this is stored in the **description** attribute. All STEP **product**s must be founded in some **product_context** which identifies the engineering discipline that is the product lead discipline.

In populating the data, the **id** or part number must be unique within the data population. This requires extra measures when the data is shared or exchanged outside of an organization.

AP 210 requires that all **product**s exist in at least three **product_category**s.

Products in AP 210 require either a **person_and_organization** or **organization** in the role of "design owner". This designation is applied to the person_and_organization or organization or design authority who designed the product.

Pre-processor Recommendations: All pre-processors should use non-defaulted data or user input for the values assigned to the design owner of a **product** as defaulting this data has a high probability of causing the data to be incorrect.

If the data is intended for external usage, or if the data repository contains product data from multiple organizations, the **organization id** attribute from the entity in the role of "design owner" shall be duplicated and pre-pended to the **product id** to provide a unique product identification**.** For example, if the **organization id** value were "COUNTRY,USA,CAGE,93699" and the part number were "999999", the actual value of the **product id** would be "COUNTRY,USA,CAGE,93699,999999".

Post-processor Recommendations: When receiving data from external sources, use the **organization id** attribute from the entity in the role of "design owner" to provide a product identification context in order to extract the data provider’s part number from the **product id.**

2.3.1.2 The Product Definition Formation Entity

AP 210 requires that all **product**s be associated with a **product_definition_formation** entity. This relation is required as AP 210 is required to support the versioning of parts. This requirement ensures that all information which typically varies from version to version is always related to the part.

There are many organizations which claim quite firmly (and possibly rightly so) that they do not version parts. All that is being done here is establishing a connection which may or may not have valuable data.

In AP 210, the connection being established is actually a connection to the data which comprises the detailed product model for the part. If your organization versions parts, the **product_definition_formation id** attribute is the standard mapping of the value which represents this version. The **description** attribute should contain the reason for the creation of the version.

AP 210 requires that all **product_definition_formation** entities be associated with a **person_and_organization** or an **organization** in the role of "creator". This is the one which created the change.

AP 210 requires that all **product_definition_formation** entities be associated with at least one **person_and_organization** or **organization** in the role of either "design supplier" or "product supplier". The person_and_organization or organization in the role of "design supplier" is the one which was the custodian of the master data when the version was created. The person_and_organization or organization in the role of "product supplier" is the one which had manufacturing cognizance (if the part is made internally to the organization) or the vendor who supplies the part if it is a vendor part.

AP 210 requires that all **product_definition_formation** entities be associated with an **approval**.

AP 210 requires that all **product_definition_formation** entities be associated with a **security_classification**.

Pre-processor Recommendations: If your organization does not version parts, the **id** attribute should contain a null string as minimal data content or any mutually agreed upon string. If the **id** attribute is a null string, the **description** value would also be a null string. All pre-processors should use non-defaulted data or user input for the values assigned to the creator, design and product suppliers, approvers and approval date for **product_definition_formation** entities as defaulting this data has a high probability of causing this data to be incorrect.

It is recommended that pre-processors use an **id** of "ANY" where they wish to indicate a generic revision of a part. This type of instancing would be used when the part with the revision of "ANY" is a component in an assembly to indicate that any existing revision of the component is valid for the assembly. This type of instancing reduces the amount of data to be sent in change packages. When this is used, it reduces the ability to track the actual contents of parts lists at a particular change level when the organization versions parts.

Post-processor Recommendations: When the value of the **id** and **description** attributes for **product_definition_formation** is a null string, post-processors should use this as an indication that there is no version of the part.

It is recommended that post-processors recognize an **id** of "ANY" as indicating a generic revision of a part. This type of instancing would be used when the part with the revision of "ANY" is a component in an assembly to indicate that any existing revision of the component is valid for the assembly.

2.3.1.3 The Product Definition Formation With Specified Source Entity

The **product_definition_formation_with_specified_source** entity is a subtype of the **product_definition_formation** entity. Therefore all guidance provided for the **product_definition_formation** entity applies to **product_definition_formation_with_specified_source.**

Pre-processor Recommendations: Pre-processors should implement the guidance for the supertype **product_definition_formation**. The **source** attribute must contain a value of ".MADE.", ".BOUGHT." or ".NOT_KNOWN.". The value should be ".MADE." if the part is built within the company. The value should be ".BOUGHT." for vendor parts. If it is possible for the part to be both made and bought (depending on serial number for example), the value should be ".NOT_KNOWN.". Only the organization that is the creator of the **product_definition_formation_with_specified_source** should modify this attribute.

Post-processor Recommendations: Processors should implement the guidance for the supertype **product_definition_formation**. Processors should preserve the value of the **source** attribute.

2.3.1.4 The Supplied Part Relationship Entity

Organizations often keep track of the relationship between the internal organization part number and the vendor part number for each of the vendor supplier parts used in products. The AP 210 Application object for this concept is Supplied_product_version. The **supplied_part_relationship** entity relates two **product** identification instances where one instance provides the vendor identification (through the **product id** attribute) and the other instance provides the organization identification. The relationship is established through two instances of **product_definition,** one instance of **product_definition_formation_with_specified_source** and one instance of **product_definition_formation**. The relationship is directed, with the 'related' value directed toward the vendor identification. The vendor identification is also using the instance of **product_definition_formation_with_specified_source**. This direction is invariant among exchange and data sharing scenarios between organization and vendor. Note that there may be more than one **supplied_part_relationship** for each vendor supplied product identification.

Pre-processor Recommendations:Vendor **product_definition_formation_with_specified_source source** attribute value should be ".MADE.", and the 'related' direction should be toward the vendor product identification.

Post-processor Recommendations:Values received should be maintained by the processor.

2.3.1.5 The Product Definition Entity

AP 210 and STEP use the **product_definition** entity to establish specific life cycle stage and design discipline views of the product data. AP 210 also uses **product_definition** and subtypes to provide a domain specific product model. The **product_definition_context** entity is used in AP 210 to distinguish view context (e.g., the design discipline identification). The AP 210 mapping table establishes the relationship between the **product_definition** entity attribute values and the corresponding context entity attribute values. In some cases, both the **product_definition_context** and the **product_context** attribute values are required to establish the context. In the cases of **shape_aspect** subtypes **part_template_definition** and **component_shape_aspect**, there are not AP 210 specific subtypes of **product_definition** required, but there are AP 210 requirements applied on the generic **product_definition** referenced by these **shape_aspect** subtypes.

It is possible to have many **product_definition**s for a part/version combination.

AP 210 provides domain specific **product_definition** subtypes in addition to the STEP generic subtypes. The following **product_definition** subtypes are considered in the AP 210 domain as complete product definitions:

- **assembly_definition,**
- **bare_die,**
- **functional_unit,**
- **electrical_network,**
- **interconnect_definition,**
- **package,**
- **packaged_connector**
- **packaged_part,**
- **physical_unit,**
- **thermal_network.**

A complete product definition in this context means that it may be the only **product_definition** in the exchange that directly points to the **product_definition_formation** that points to the **product** about which information is being exchanged.

The following "composing" **product_definition** subtypes are considered in the AP 210 domain as product definitions that are used in composing a complete product definition:

- **bus_structural_definition,**
- **component_definition,**
- **component_functional_unit,**
- **network_node_definition,**
- **packaged_component**
- **printed_component,**
- **stratum**
- **thermal_component.**

The "composing" **product_definitions** in this context implies that the **product_definition** is not required to point to a real **product**, (although it may point to a real **product)**.

AP 210 provides the following externally defined **product_definition** subtypes to further establish the context:

- **externally_defined_assembly_definition,**
- **externally_defined_bare_die,**
- **externally_defined_functional_unit,**
- **externally_defined_interconnect_definition,**
- **externally_defined_package,**
- **externally_defined_packaged_connector,**
- **externally_defined_packaged_part,**
- **externally_defined_physical_unit.**

The above items are subtypes of both **product_definition** and **externally_defined_item**.

AP 210 provides the following externally defined **product_definition** subtypes to further establish the context:

- **library_defined_assembly_definition,**
- **library_defined_bare_die,**
- **library_defined_functional_unit,**
- **library_defined_interconnect_definition,**
- **library_defined_model,**
- **library_defined_package,**
- **library_defined_packaged_connector,**
- **library_defined_packaged_part,**
- **library_defined_physical_unit.**

The above items are subtypes of both **product_definition** and **externally_defined_item** with a further requirement that **external_source** attribute **source_id** is the library identification. In the AP 210 domain, libraries are not versioned, only the items in the library are controlled. The **library_defined_model** is additionally a subtype of **analytical_model**. This provides the functionality to provide the same level of configuration management to simulation models as available to product configuration management. All simulation model configuration management should use this subtype.

The shape_aspect entity provides subtypes that are used for definitions in the domain of interconnect and that use **product_definition** and **product_definition_context** as supporting entities. Important concepts such as land templates and other library symbology are supported in this manner. Instance data in the domain of interconnect design is also supported with subtypes of **shape_aspect**.

2.3.1.5.1 The Product_definition attributes

The **id** and **description** attributes of **product_definition** are used to identify either "type" or "instance" values. In this context "type" refers to a generic concept similar to an EXPRESS entity where the interpretation process did not create an AP 210 specific entity but chose to encode the information as a string value. In this context "instance" refers to the more usual identifier role. The "type" values are directly related to the requirements specified in relevant Application objects in clause 4 of AP 210.

The following are standard "type" mappings for the **id** attribute:

- "design composition path",
- "reference composition path".

The following are standard "type" mappings for the **description** attribute:

- "altered package",
- "altered packaged part",
- "bare die component",
- "join 2 physical connectivity definition supporting printed component",
- "laminate component",
- "mating connector"
- "packaged connector component",
- "placement group",
- "printed connector component",
- "routed packaged component",
- "thermal component".

These mappings are in the limited context as established by **product_definition_context** as documented in the AP 210 mapping table for the Application objects that correspond to these type names. The reference path and associated constraints define the structure that establishes the valid context.

Note: The **part_template_definition** is closely aligned with this usage but since it a subtype of **shape_aspect** it is not discussed above. The **id** and **description** attributes of the **product_definition** associated with the **part_template_definition** are not directly controlled since the **product_definition_context** provides an unambiguous context for **part_template_definition** instance data.

The following are standard "instance" mappings for the **id** attribute:

- reference designation for a functional unit in a hierarchical netlist,
- stratum name,
- mating connector component reference designation.

The following are standard "instance" mappings for the **description** attribute:

- context description of AP 210 application object Ee_product_definition and subtypes,
- "primary design layer stratum",
- "non primary design layer stratum",

The functional unit reference designation is a string usually composed of a concatenation of a function type and an integer in the limited context of functional networks as established by **product_definition_context**. Functions may be categorized into types using the **product_category** entity. Reference the AP 210 mapping table for the Application object Functional_unit for the exact structure that establishes the valid context.

The stratum name is usually represented by an integer (which indicates a relative position). Note that this integer is not the standard method for formally establishing the position of a stratum in the board stackup. There are no facilities in AP 210 for ensuring the relevant properties of the stratum name necessary for the integer to be an accurate representation of the position in the stackup. The only requirement in AP 210 on the stratum name is that it be unique for each instance of stratum. AP 210 uses **shape_aspect_relationship** between adjacent stratum surfaces to ensure unambiguous description of the stratum position in the stackup. The **shape_aspect_relationship** is directed; the stratum surfaces are identified as either primary or seconday; the result is that the stackup is determinant without relying on descriptors. The description attribute on **shape_aspect_relationship** shall be 'adjacent stratum surface definition'. The Inter_stratum_extent Application object should not be used to define stratum stackups, although there are some obvious business rules for valid Inter_stratum_extents (i.e., the extent has to exist!).

Organizations occasionally require explicit definition of mating connector relationships. This is modelled in AP 210 using the The Application object Mating_connector_component. A reference designation is usually composed of a concatenation of a type associated with a connector and an integer and is only applied to a **product_definition** of the type **component_definition** with a **description** value of "mating connector".

AP 210 requires that all **product_definition**s have a **person_and_organization** assigned in the role of "creator". This person_and_organization or organization is the one which defined the view. If the **product_definition** is being used as solely a connection to a property, this would be the person who filed the computer model of the property.

AP 210 requires that all **product_definition**s have a **date_and_time** assigned in the role of "creation date". This date and time is when the view was defined. If the **product_definition** is being used as solely a connection to a property, this would be the file date and time for the computer model of the property. If this is not the case, see the pre-processor recommendations.

AP 210 requires that all **product_definition**s have an **approval**. If this data is difficult to determine, use the creator data as source data.

Pre-processor Recommendations: Since there are so many standard mappings on the id and description attribute, it is infeasible to use the id attribute for describing the engineering discipline that created the definition. See the **product_definition_context** entity for the discipline.

Where values for the creator, creation date are not readily available, this information can be extrapolated from the creator and approval related to the **product_definition_formation**. Pre-processors shall not use **product_definition_with_associated_documents** to relate specification type documents to the **product_definition**.

Post-processor Recommendations: Post-processors shall follow the requirements of the mapping tables.

2.3.1.6 The Product Definition With Associated Document Entity

AP 210 has an optional feature where a **product_definition** may be related to **document** entities through the sub-type **product_definition_with_associated_documents**. In AP 210, this usage is intended only for documents which identify associated computer files where the **document_type** attribute **product_data_type** has the value "cad filename". A non-standard usage would be to associate computer files for drawings where the **document_type** attribute **product_data_type** has the value "drawing". The non-standard usage would require an exchange agreement.

There is a rule in AP 210 restricting the **product_data_type** of **ee_specification** subtype of **document**s, but the AP does not formally eliminate the possibility of relating **ee_specification** type **document**s using the **product_definition_with_associated_documents** sub-type.

When using **product_definition_with_associated_documents** to reference computer files, the **document id** attribute should contain the file name of the file with enough detail so that it is uniquely identified in the exchange. The following is the recommended list of {type, value} tuples:

- "OS_SUPPLIER_ID",
- "string",
- OS_NAME",
- "string",
- "OS_VERSION_ID",
- "string",
- "CAD_TOOL_SUPPLIER_ID",
- "string",
- "CAD_TOOL_NAME",
- "string",
- "CAD_TOOL_VERSION_ID",
- "string",
- "CAD_FILE_TYPE",
- "string",
- "CAD_FILE_TYPE_VERSION_ID",
- "string",
- "CAD_FILE_LOCATOR_TYPE".
- {"URL" | "DIRECTORY PATH"},
- "string",
- "CAD_FILE_DATE_TIME_STAMP",
- "string".

All uppercase strings should be provided verbatim, with the actual parameter values included in the positions identified by the keyword 'string' in the above list. This list is self-explanatory except for:

- "OS_SUPPLIER_ID" and "CAD_TOOL_SUPPLIER_ID" should be coded in accordance with **id** attribute recommendations for the AP 210 entity **organization**.
- "CAD_FILE_TYPE", which is the language reference manual for the file type or standard (e.g. IGES, STEP AP schema name, POSTSCRIPT, etc.));
- "CAD_FILE_TYPE_VERSION_ID", which is the version of the reference for the file type;
- "CAD_FILE_LOCATOR_TYPE", which is the first element of a three element tuple consisting of it plus an indicator of either a URL or an internal directory path, plus the value of the URL or internal directory path.
- "CAD_FILE_DATE_TIME_STAMP" should be coded in accordance with AP 210 date_and_time entity coding requirements, in order to be consident with the rest of the standard.

Due to the coding of data in the **id** attribute, the use of the **name** attribute and the **description** attribute is unnecessary.

Pre-processor Recommendations: Reference **product_definition_context** for discipline and life cycle view attributes. In order to implement concurrent engineering it is expected that exchange agreements would need to be implemented. Where values for the creator, creation date are not readily available, this information can be extrapolated from the creator and approval related to the **product_definition_formation_with_specified_source**. Pre-processors shall not use **product_definition_with_associated_documents** to relate specification type documents to the **product_definition**.

Pre-processors should implement the standard mappings as discussed above.

Pre-processors should use the coding specified in this document as computer sensible identification of sending system configuration for the **document id** attribute.

Post-processor Recommendations: All post-processors shall implement the standard mappings as discussed above.

All post-processors should utilize the values given above for **document id** attribute for pre-processors or implement appropriate exchange agreements.

2.3.1.7 Shape as a property

Shape is a basic property supported in AP 210 explicitly. Because it is possible to have several analysis shapes for the same product, the representation that identifies the purpose and other state information for the shape effectively identify the state of the product definition for which that shape is a property. This is required so that the alternate representation construct needed for backward compatibility with AP 203 can be supported. Reference clause 1.5.4.1 in this document for a description of the alternate representation recommendations.

2.3.1.8 Shape Aspect Entity

The **shape_aspect name** and **description** attributes standard mappings are defined in the mapping tables, clause 5.1 in the standard. There are many AP 210 domain specific subtypes of **shape_aspect**. Two subtypes of **shape_aspect** are of particular importance: **part_template_definition** and **component_shape_aspect**. **part_template_definition** is supported by a **product_definition** in a context of 'template definition' and is a definitional concept itself. **component_shape_aspect** is the occurrence of a **part_template_definition**, so there is an 'occurrence' context for instances of this entity type. There is a **shape_aspect_relationship** of 'instantiated template' identifying the definition for a **component_shape_aspect**.

2.3.1.9 Categorizing Products

AP 210 provides for assigning parts to categories and for creating a network of categories. There are standard categories in the AP 210 domain used to establish exchange contexts. Category networks can be extremely useful in adding intelligence to the data. The use of a network graph rather than a tree is crucial to the usefulness of the approach. A tree based approach with a single root node is impractical. AP 210 supports exchange of data sufficient to support part selection processes. Parts are assigned directly to categories though the **product_related_product_category** entity which is a sub-type of the **product_category** entity. Category hierarchies can be created by relating super and subcategories through **product_category_relationship** entity. An AP 210 product could be related to as few as one root node or as many root nodes are of interest in the enterprises exchanging or sharing data. In this way, multiple characteristics may be related to a product (e.g., "commercial", "serialized", "replaceable", "standard part", "assembly"). In this example, "assembly" is the value assigned to **product_related_product_category name** attribute. The other strings would be contained in individual instances of **product_category name** attribute and related to the **product_related_product_category** pointing to this **product** by four (4) instances of **product_category_relationshp**.

Pre-processor Recommendations: There are no standard mappings for the **product_category_relationship name** or **description** attributes. Since there are no standard mappings in the AP 210 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate/ mutually agreed upon string. Pre-processors should use lower case for **product_category name** values. Leading and trailing blanks in the **product_category name** value should be removed.

Post-processor Recommendations: Since there are no standard mappings for the **product_category_relationship name** or **description** attributes, it is recommended that post-processors not assign any processing significance to these values. Post-processors should attempt to store all categories and subcategories and category relationships received in an AP 210 exchange as this information adds meaning to the received data. If it is impossible to store the data, the user should be informed of all categories and relationships not processed. This would be best done by presenting the user with a report on the category structure in the file with subcategories indented. Post-processors should use non-case sensitive checking when determining matches on processed category data. Leading and trailing blanks in the **product_category name** value should be removed.

2.3.1.10 AP 210 Standard Categories

AP 210 has the capability to exchange network structures of product categories where certain nodes in the network are standardized in the AP 210 domain. AP 210 rules control network entries that directly reference **product**s. Generic names are included for interoperability purposes in order to allow product identification. The **name** attribute of each instance of **product_related_product_category** is constrained to be one of the following values:

"assembly",

"assembly module",

"bare die",

"cast",

"coined",

"customer furnished equipment",

"design layer",

"detail",

"documentation layer",

"drawn",

"extruded",

"forged",

"formed",

"functionality",

"inseparable assembly",

"interconnect",

"interconnect module",

"machined",

"material",

"molded",

"package",

"packaged part",

"part",

"piece part",

"printed part",

"raw material",

"requirements model",

"rolled",

"sheared",

"simulation model",

"standard",

"template model",

"tool".

These categories should be considered as basic product types. If a network of **product_related_product_categories** that reference a **product** or **products** contains an instance of the **product_related_product_category** that is a "standard" category, the number of **product_related_product_categories** that are in that network shall be one or two. Otherwise the number of **product_related_product_categories** that reference a **product** or **products** shall be one. This mechanism helps to constrain the context for instances of **product_definition** (and its subtypes) and **shape_aspect**(and its subtypes).

As an example use of this capability, an interconnect substrate may be a flex rigid pcb. Normal industry usage for the network structure creation would be for the string "pcb" to be assigned to a higher level category **name** attribute, and each of the strings "flex-rigid", "flex", "rigid" to be assigned to a lower level element in the category network. One of the nodes (e.g., "flex-rigid") would be related to the **product_related_product_category** by a **product_category_relationship** entity.

If the **product_related_product_category** assigned to the specific product under design or usage had the string value "interconnect module" assigned to the **name** attribute, that would have the effect of applying an industry specific or enterprise specific classification ("flex-rigid") to the AP 210 standard category of "interconnect module".

Industry specifications and standards (e.g., IEC, IPC, ARINC) should be used where possible to establish the product category networks to maximize the benefit of this capability. It is recommended to represent the specification identifier (e.g., "IEC 1182-10") in the **product_category description** attribute, since there is no standard mapping or character strings assigned to this attribute in the AP 210 domain.

There are a few pre-defined strings in AP 210 for the value of the **product_category name** attribute when the instance created is not a **product_related_product_category**. The meanings for these pre-defined strings are included below. Otherwise there are no restrictions on the attribute. This means that when networks of categories are created there is no restriction (other than that stated above) on the names of the categories not directly related to a **product**.

Not all products in AP 210 are directly physically realizable (i.e., functionality, simulation model, requirements model, printed part, technology specific land pattern, technology specific padstack, template model).

2.3.1.11 Standard Parts

AP 210 defines via its mapping table a mapping for standard parts. The defined mapping is that a **product_related_product_category** have the standard character string assigned of "standard". This mapping is in agreement with the PDM schema at this time. This mapping is different than the mapping of the DIS version of the standard.

2.3.1.12 Technology Specific Padstack

Some 2D CAD systems define default padstacks. AP 210 separates the (conceptual) definition of the barrel from the definition of the associated lands. Since a particular padstack is always based on a specific stratum stackup, a specific use of Interconnect_module is recommended that is composed of the specific Lands and Stratum needed to support the padstack definition. The recommendation is that a **product_category** be created and that this category be related through a **product_category_relationship** to a **product_related_product_category** related to a **product**. The recommended character string assigned for this purpose is "technology specific padstack". This category should only exist if the "interconnect module" **product_related_product_category** is the relating end of the **product_category_relationship**.

2.3.1.13 Technology Specific Land Pattern

IPC Standards provide reference land patterns. These patterns are purely geometric shapes and have no connectivity information. AP 210 provides no explicit way to define land patterns because the location of the land in the pattern is derived from the connection_zone location for the terminal in the Package model. A Part_template is intended to be an individual thing. The standard is very specific in its use of Assembly_joint to ensure that the connectivity defined in the netlist is implemented in the Interconnect_module. In order to allow support for land patterns and subsequently derive footprints a specific use of Interconnect_module is recommended that is composed of the Lands necessary to support a specific Packaged_part. The recommendation is that a **product_category** be created and that this category be related through a **product_category_relationship** to a **product_related_product_category** related to a **product**. The recommended character string assigned for this purpose is "technology specific land pattern". This category should only exist if the "interconnect module" **product_related_product_category** is the relating end of the **product_category_relationship**. This product may include padstacks through the use of "technology specific padstack" products and breakout patterns and micro-vias etc.

2.3.1.14 Reference Packaged Part Assembly Implementation

AP 210 supports gathering the information related to a specific combination of Packaged_part, assembly technology, and fabrication technology together in one place. The method is to use the Assembly_module as the composition vehicle. This provides a vehicle to gather together the bond definition, the component definition and the interconnect definition. The recommendation is that a **product_category** be created and that this category be related through a **product_category_relationship** to a **product_related_product_category** related to a **product**. The recommended character string assigned for this purpose is "reference packaged part assembly implementation". This category should only exist if the "assembly module" **product_related_product_category** is the relating end of the **product_category_relationship**.

2.3.1.15 Reference Packaged Part Interconnect Implementation

In order to support the particular fabrication technology, land patterns, padstacks, breakouts, and material stackup for a "reference packaged part assembly implementation", a particular Interconnect_module must be defined, since an Interconnect_module requirements are dependent on the requirements driven by the assembly. The reference interconnect may be composed of technology specific land pattern, technology specific padstacks, and other needed interconnect elements necessary to form a valid interconnect definition. The recommendation is that a **product_category** be created and that this category be related through a **product_category_relationship** to a **product_related_product_category** related to a **product**. The recommended character string

assigned for this purpose is "reference packaged part interconnect implementation". This category should only exist if the "interconnect module" **product_related_product_category** is the relating end of the **product_category_relationship**.

2.3.1.16 Recommended Categories

It is recommended that all implementations of AP 210 support the following high level categories which are not standardized in the AP, but will undoubtedly have common usage:

- "commercial" - This category indicates that the **product** referenced by a product_related_product_category that is related to this category (by a product_category_relationship) is a general-purpose commercially-available product.
- "customer furnished customer installed" - This category indicates that the **product** referenced by a **product_related_product_category** that is related to this category (by a **product_category_relationship**) is part of the system or unit for requirements definition, but is actually placed in the system or unit after some portion of delivery.
- "government" - This category indicates that the **product** referenced by a **product_related_product_category** that is related to this category (by a **product_category_relationship**) is a product which has been developed or purchased to meet specialized government specifications.
- "hazardous material" - This category indicates that the **product** referenced by a **product_related_product_category** that is related to this category (by a **product_category_relationship**) either is or contains hazardous material.
- "interchangeable" - This category indicates that the **product** referenced by a **product_related_product_category** that is related to this category (by a **product_category_relationship**) is a product which requires no trimming or modification when replaced.
- "replaceable" - This category indicates that the **product** referenced by a **product_related_product_category** that is related to this category (by a **product_category_relationship**) is a product which requires trimming or some modification (usually for fit) when replaced.
- "serialized" - This category indicates that the **product** referenced by a **product_related_product_category** that is related to this category (by a **product_category_relationship**) is (or contains) a serialized product.

If multiple classifications are desired (for instance "commercial" and "replaceable"), then multiple instances of **product_category_relationship** should all point to the same instance of **product_related_product_category**. If desired, these could all be sub-categories of a product_category with a name attribute of "contractual classification".

2.3.2 Relating Specifications to Products.

AP 210 provides backward compatiblity to AP 203. Using the AP 203 method, one relates specifications to entire parts by relating an applied_document_reference entity specified by applied_document_reference to the product and by the classification of the document as a specification. If the specification only relates to a portion of the part, the applied_document_reference document is related to a shape_aspect which is in turn related to the product_definition_shape of the part. AP 210 provides the ability to relate documents to product_definition_formation, product_definition and several other entities using the same mechanism.

The applied_document_reference entity identifies the owner of the specification through the source attribute. This attribute should contain an unambiguous identification of where the receiver of the data could obtain a copy of the document. The document related to the applied_document_reference must be uniquely identified in the exchange by the id attribute. This means that the id should contain any revision information needed to identify the document completely. The name attribute should contain the title of the document. The description attribute should contain an expanded explanation of the document's contents.

Since many specifications cover a variety of subtopics and options on a given topic, it may be necessary to identify a particular subtopic of the specification and assign option values. In AP 210 this is accomplished by relating a document_usage_constraint to the document. The subject_element attribute identifies the particular section or topic being referenced in the specification. The subject_element_value identifies any option choices or restrictions placed on the section or subtopic.

The document_usage_constraint should not be used to reference classes defined in specifications such as process specifications. This should be done by using the document entity sub-type document_with_class. If classed documents require further restriction of the class, a document_usage_constraint may be related to the document_with_class entity. This is also provided for backward compatibility. See the above note.

AP 210 provides for documents related to a product_definition to be related to other documents in a network type relationship. This is accomplished through the document_relationship entity. The standard usage for this relationship is to describe a version of a document. Reference clause 5.1 in AP 210.

AP 210 provides for directly exchanging requirements allocated to a product where the requirements were extracted from a document and where there may be more data than can conventiently fit in the document_usage_constraint. Reference requirements section of this document.

Post-processor Recommendations: Post-processors should store all data found in specification documents attached to product_definitions or shape_aspects. If it is not possible to store all the data, the user must be informed of the data being omitted and its relationship to the product_definition or shape_aspect.

2.3.3 Relating Product Data to Contracts

AP 210 provides an optional relationship of the following entities:

Products,

Product_definition_formation,

Alternate_product_relationship,

Directed_action to contracts through the applied_contract_assignment entity.

In AP 210, a contract can be used to represent either an explicit contract which provides the requirements (and typically the funds) for the activity or some other agreement (such as a purchase order) which fulfills the same function. The contract name attribute should contain the contract or agreement identifying number or name if no number exists. The purpose attribute should contain the reason for the existence of the contract or agreement. The contract_type description attribute may contain values such as "fixed_price" or "cost_plus" or other descriptors.

AP 210 requires that a contract have an associated approval. A contract must also have an associated person_and_organization in the role of "contractor". A contract may have an associated date_and_time in the role of "contract_date".

Pre-processor Recommendations: It may be difficult to obtain the approval and contractor information. If this information is not available, it should be provided either through user input or from default data based on the contract name value.

2.3.4 Relating Properties to Products

2.3.4.1 Shape State Information for product usage views

The AP 210 reference CAD model for the usage view [1]includes the following properties in the ARM Application objects Physical_unit_planar_shape and Physical_unit_3d_shape to distinguish purposes for creating shape data:

- application_technology_constraint,
- shape_material_condition,
- centroid_location,
- shape_environment,
- shape_purpose.

The shape_purpose is one of:

- thermal_analysis,
- vibration_analysis,
- shock_analysis,
- electromagnetic_compatibility_analysis,
- design,
- design_profile,
- design_profile_above_seating_plane,
- design_profile_below_seating_plane.

The shape_material_condition is either: nominal_material_condition, maximum_material_condition, least_material_condition.

The shape_environment is either: end_user_application, manufacturing.

2.3.4.2 Assigning Shape properties to products

For backward compatibility to AP203, and specifically in support of shape properties, AP 210 uses several entities to form the link between the configuration management data for a product and the properties for a product. These entities are **property_definition, property_definition_representation, product_definition_shape** and **shape_definition_representation**. There are no standard mappings for the product_definition_shape name and description attributes. It should be noted that no link to **property representation** (such as shape) is required. It is possible to use the **property_definition** or **product_definition_shape** entity to indicate that a product has (or will have) properties without relating a **property_definition_representation** or **shape_definition_representation**.

There must be only one **product_definition_shape** for each **product_definition** in an AP 210 exchange file. If there are multiple **shape_definition_representation** entities related to the

1. Reference AP 210 Concept of Operations (1999)

product_definition_shape, these relationships describe alternate representations. This is depicted in Figure 3.

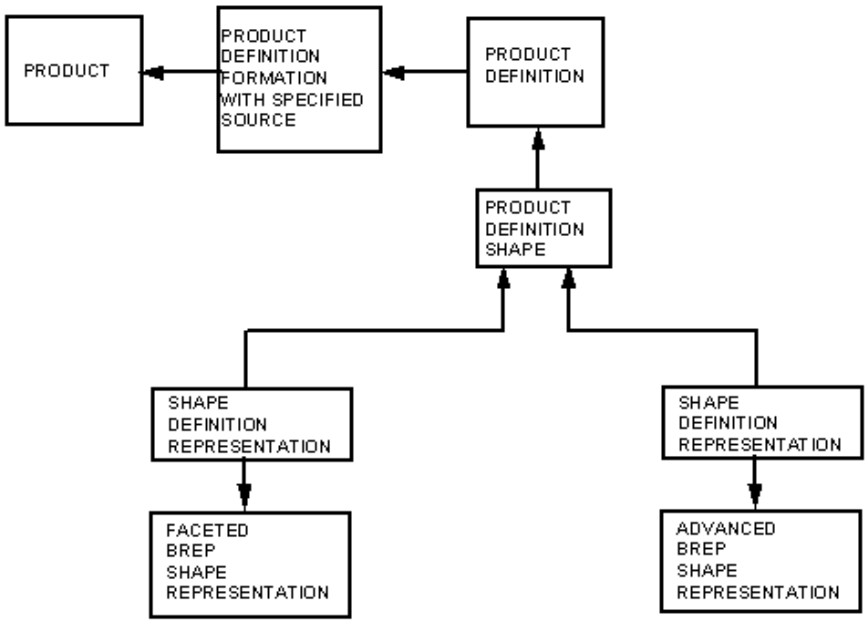


FIGURE 3 PRODUCT WITH ALTERNATE SHAPE REPRESENTATIONS

If the shape of the product is composed of shape constructs from multiple types of shape_representation to form the entire shape model, the main shape_representation shall be related to a shape_definition_representation which relates to the product_definition_shape. The other

shape_representations are related to the main shape_representation through a shape_representation_relationship. This is depicted in Figure 4.

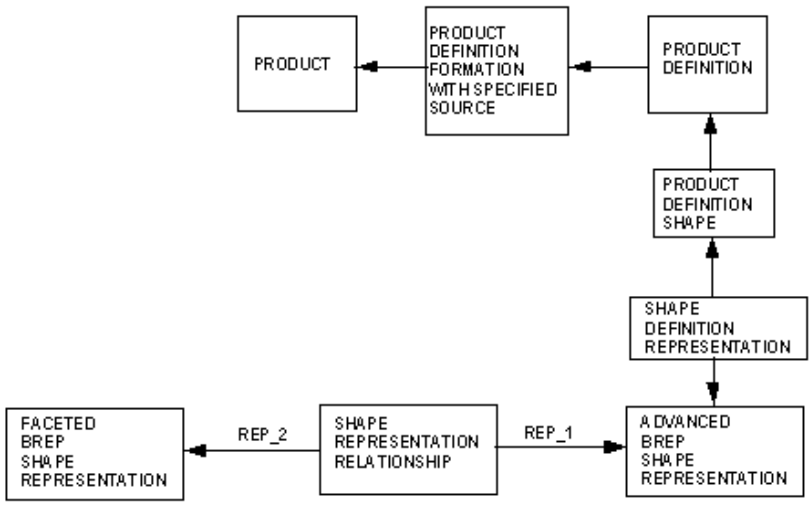


FIGURE 4 SINGLE PRODUCT WITH MULTIPLE SHAPE REPRESENTATIONS

In some cases, the shape of a product is based on the shape of another product. This commonly occurs when one is the mirror image of the other. When this occurs, it is through a representation_relationship_with_transformation. This is structure is shown pictorially in Figure 5.

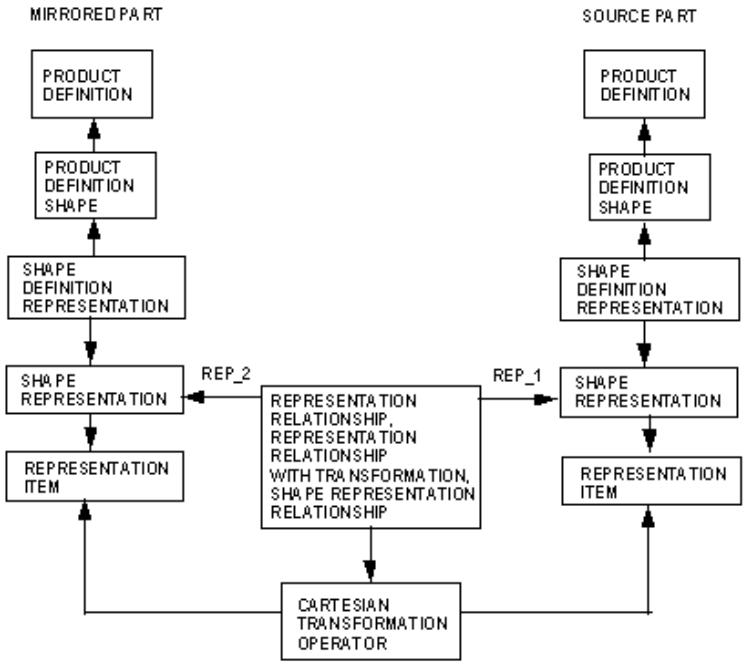


FIGURE 5 PRODUCT SHAPE DERIVED BY MIRRORING ANOTHER PRODUCT SHAPE

The transformation is constructed based on a functionally_defined_transformation. It is presumed that the transformation would be applied to the coordinate system of the source part prior to it being mapped to that of the mirrored part.

Pre-processor Recommendations: There are no standard mappings for the name and description attributes for product_definition_shape. Since there are no standard mappings in the AP 210 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate/mutually agreed upon string.

Post-processor Recommendations: Since there are no standard mappings for the name and description attributes for product_definition_shape, it is recommended that post-processors not assign any processing significance to these values.

2.3.4.3 Assigning Non-shape properties to products

For support of product properties applied at the product level only, and not dependent on product versions, AP 210 provides property assignment through specific subtypes of the **product_category** and **characterized_object** entities. Reference the section in this document specifying product categories for category determination. The specific AP 210 entities are **characterized_product_category, product_related_characterized_product_category**. A **characterized_product_category** may have one or more properties associated with it. The properties may be tuples of (identifier, data) or may be a data element type name assignment only. A **product_related_characterized_product_category** then associates a specific property assignment value for a specificy data element type for a specific product. For example a **characterized_product_category** may be a RAM memory array. There may be a memory size property assigned to the **characterized_product_category** as a data element type but no value assigned at the **characterized_product_category** level. A specific product would have a **product_related_characterized_product_category** with an assignment of "memory size" of "128Mb" as a value. The property "memory size" would be inherited from the **characterized_product_category** as an allowed data element type. Properties not included in a **characterized_product_category** are not allowed to be assigned values in the **product_related_characterized_product_category**.

In the AP 210 domain, it is not required that IP1 on **product_category** or IP1 on **product_category_relationship** be satisfied. AP 210 allows properties to be assigned to the AP 210 specific subtypes that are also subtypes of **characterized_object**. The assignment of specific property values requires multiple instances of entities which cause the violation of the two indicated IPs.

2.3.5 Renumbering Vendor Parts

In all realms of design and manufacturing business, it is common to buy parts from a vendor and renumber them under an internal numbering scheme. In today's practice, this is done through envelope, specification and source control drawings. An envelope drawing is used for a simple renumber of a part where the part is referenced on the envelope drawing and assigned a new part number via the associated parts list. A specification control drawing renumbers a part to show that it meets or exceeds the specifications defined on the drawing and to recommend sources for the part. A source control drawing renumbers a part and creates a restricted list of suppliers which are qualified to produce the part based on the specifications..

In AP 210, all of the above relationships are supported through the supplied_part_relationship. This relationship is used for the identification of part_suppliers and design_suppliers. The identification of "design_supplier" is actually redundant, since this information can be obtained from the person_and_organization related to the product_definition_formation in the role of "design_supplier". This document will only address the use of supplied_part_relationships for renumbering of parts. The structure of a supplied_part_relationship is shown in Figure 6.

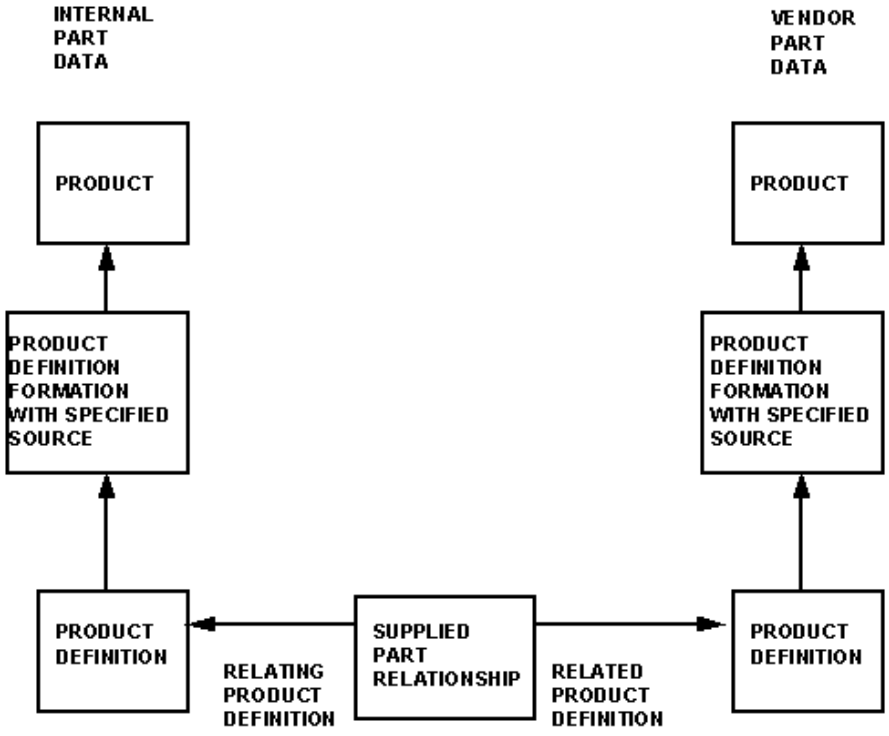


FIGURE 6 SUPPLIED PART RELATIONSHIP

To renumber parts through a supplied_part_relationship, both parts must be defined. The supplied_part_relationship relates the "customer" part number's product_definition in the relating_product_definition attribute to the "supplier" part number's product_definition in the related_product_definition attribute. There are no standard data or mappings for the name and description attributes. The id attribute must be unique, but there is, again, no standard mapping.

Certification of suppliers can be indicated through a supplied_part_relationship. This is accomplished by populataing an applied_certification_assignment which relates a certification to the relationship. There are no standard mappings for the values of the name and purpose attributes for the certification entity. There are no standard mappings for the values of the certification_type description attribute. If a certification is used, AP 210 requires that the certification be related to an approval. It is further required that the certification be associated with a date_and_time in the role of "certification_date".

Pre-processor Recommendations: It may be difficult to obtain the data for the certification's approval and "certification_date". Where this data is not immediately available, it can be extrapolated from the approval related to the **product_definition_formation** found on the path referenced by the relating_product_definition attribute.

There are no standard mappings for the name and description attributes in a supplied_part_relationship. Since there are no standard mappings in the AP 210 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate/ mutually agreed upon string. The id attribute must be constructed so as not to duplicate any assignments made to other entities which are sub-types of product_definition_relationship.

There are no standard mappings for the name and purpose attributes in a certification. Since there are no standard mappings in the AP 210 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate/ mutually agreed upon string.

Post-processor Recommendations: Since there are no standard mappings for the name and description attributes for a supplied_part_relationship, it is recommended that post-processors not assign any processing significance to these values.

Since there are no standard mappings for the name and purpose attributes for a certification, it is recommended that post-processors not assign any processing significance to these values.

2.3.6 Alternate Part Concepts

AP 210 supports several variants of designating alternative parts.

2.3.6.1 Alternate Parts

Alternate parts in AP 210 are defined through the **alternate_product_relationship** entity with the **basis** attribute assigned a value of 'alternate product'. This relationship is used in the definition of parts list data for alternate item designations. There are no standard mappings to the name and description attributes of this entity.

Pre-processor Recommendations: There are no standard mappings for the name and description attributes of alternate_product_relationship. Since there are no standard mappings in the AP 210 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate/ mutually agreed upon string.

Post-processor Recommendations: Since there are no standard mappings for the name and description attributes of alternate_product_relationship, it is recommended that post-processors not assign any processing significance to these values.

2.3.6.2 Test Select Parts

AP 210 provides support for exchanging one or more lists of parts that may be selected from at final test. The list is identified as a single item on the parts list for the product, but the quantity of each item in the list is an indication of the relative frequency of occurrence of that value in production of the end product. The standard mapping for this is **alternate_product_relationship basis** attribute is 'test select product'. The list is a product with it's own part number, version, and parts list. The recommendation is that the relative frequency of occurrence is normalized to a total population of 100, with 2 digit precison applied. This allows the use of integers where the total value of all entries in the list is 100.

2.3.6.3 Assembly specific alternate parts

AP 210 provides the ability to identify two-way alternates within the context of a single assembly by assigning the value of 'assembly alternate product' to the **basis** attribute of **alternate_product_relationship**.

2.3.6.4 Substitute parts

AP 210 provides the ability to substitute one part for another within the context of a single assembly or a single assembly relationship. The **assembly_component_usage_substitute** entity provides this capability. This is a one way substitution capability with no guarantee of form, fit, or functional 100% equivalence.

2.3.7 Product association

AP 210 provides the ability to associate two products based on a specification. This capability is encoded in the **product_definition_formation_relationship** with the **name** attribute assigned a value of 'product association'. The business purpose should be described in the specification.

2.3.8 Make From Relationships

In AP 210, the fact that the design or design usage for a product is made from the design or design usage for another product is indicated by the **design_make_from_relationship**. To indicate the above, both products must be defined in a design or design usage view. The **design_make_from_relationship** relates the source product's **product_definition** in the **relating_product_definition** attribute to the resultant product's **product_definition** in the **related_product_definition** attribute. Figure 7 illustrates this relationship.

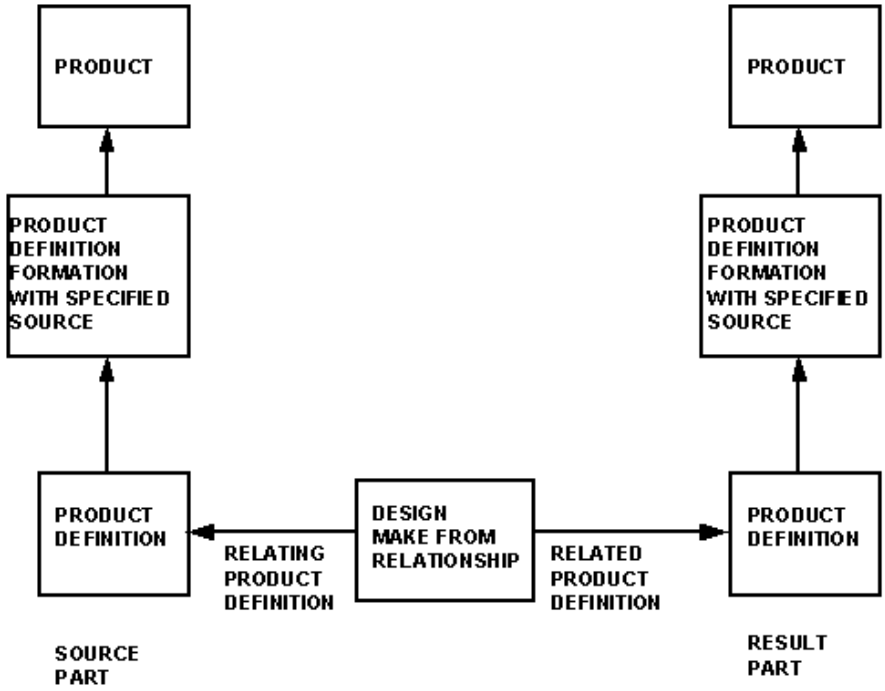


FIGURE 7 Design make from relationship

Typical usage of this capability in the electronic domain is for programmable devices.

Note: Since AP 210 supports functional products this relationship can be used to define functional definitions that are derived from other functional definitions.

Pre-processor Recommendations: The attributes should be set to null strings.

Post-processor Recommendations: It is recommended that post-processors not assign any processing significance to the attribute values for **design_make_from_relationship**.

2.3.9 Assembling Products

In AP 210, an assembly is defined as a product that has other products related to it which represent the detail products and sub-assemblies which comprise the assembly. This relationship of an assembly product to its components is defined through a **next_assembly_usage_occurrence** in AP 210. The structure of this relationship is shown in the Assembly module section of this document. Most usages of assemblies using **next_assembly_usage_occurrence** are single level bill of material. Special domain specific sub-assemblies are supported. Reference the Associated parts section of this document.

NOTE - Processors may use a version id of "ANY" where they wish to indicate a generic revision of a product when the product is a component in an assembly. This indicates that any existing revision of the component is valid for the assembly. This type of instancing reduces the amount of data to be sent in change packages. When this is used, it reduces the ability to track the actual contents of parts lists at a particular change level when the organization versions products.

The **next_assembly_usage_occurrence id** attribute inherited from **product_definition_relationship** has standard mappings defined in the mapping table. The **name** attribute inherited from **product_definition_relationship** has standard mappings defined in the mapping table. The item/find number from the parts list should be represented by a **descriptive_representation_item** with a **name** attribute value of 'item find number' and the **description** attribute of **descriptive_representation_item** assigned the value for the string representing the actual item/find number. It is recommended that the **representation_context context_type** attribute for the **descriptive_representation_item** be set equal to 'item find number representation context'. It is recommended that the **name** attribute of **property_definition** be set equal to 'item find number' as well. It is usually the case that the number is a positive integer greater than zero, but there are no rules to formalize this. It is possible and likely that multiple instances of **next_assembly_usage_occurrence** will be assigned to the same instance of **descriptive_representation_item**. The **description** attribute inherited from **product_definition_relationship** has standard mappings defined in the mapping table. The **reference_designator** attribute of **next_assembly_usage_occurrence** inherited from **assembly_component_usage** designates a unique positional location within the context of the assembly referenced by the relating_product_definition inherited from **product_definition_relationship**. There is a rule in the standard that requires the combination of **reference_designator** and **relating_product_definition** to be unique.

It should be noted that since the usage is described by a product_definition_relationship, many different views of the usage can be established by varying the relating_product_definition. AP 210 can maintain one usage based on the "design" product_definition and another based on the "manufacturing" product_definition. The various product_definitions can move into other life cycle stages for the product as well. In this way, usages or parts lists can be defined for any of a number of views and life cycle stages of a design.

AP 210 requires that all assembly_component_usages and therefore all next_assembly_usage_occurrence entities be associated with a security_classification.

Pre-processor Recommendations: The id attribute of the next_assembly_usage_occurrence must be constructed so as not to duplicate any assignments made to other entities which are sub-types of product_definition_relationship.

The security_classification classification officer, classification date, approvers and approval dates can be extrapolated from the version creator and approval data for the assembly part if no appropriate data is available.

2.3.10 Instances in Multi-Level Assemblies

2.3.10.1 Reference designation

AP 210 supports data in conformance with IEC 750:1983 through the use of reference designation for components. Use of AP 210 consistent with this IEC standard provides the ability to unambiguously identify individual component locations in the product definition. The use of reference designation often makes other forms of instance tracking through multiple levels of assembly unnecessary.

2.3.10.2 Quantities in Assemblies

AP 210 provides for designating quantities of components in next assemblies and higher assemblies. The most common types of quantities are next assembly quantity and quantity for an end item. A next assembly quantity is the amount (count or other measure) of a part in its immediate parent part. The quantity for an end item is the amount (count or other measure) of a part in a finished manufactured item. The end item itself is designated by the organization and may be a configuration item. These two types of quantity and their related data is typically what comprises the body of a parts list. Much of the normal industrial usage of AP 210 is for the end item and the next assembly to be the same instance.

2.3.10.3 Next Assembly Quantity

AP 210 provides two methods for specifying next assembly quantity. One method is to count the number of next_assembly_usage_occurrences where the pair of the relating_product_definition and related_product_definition attributes are identical among multiple instances of the next_assembly_usage_occurrence entity. This type of quantity specification can only be used for items which are counted one piece at a time as there can be no unit of measure attached to this type of quantity. This method is extremely valuable where all instances of a component are specified geometrically as well as in the product structure. This method is used where the reference designation is applied in accordance with the principles of IEC 750:1983.

The other method of specifying next assembly quantity in AP 210 is by creating a complex instance involving both next_assembly_usage_occurrence and quantified_assembly_component_usage. The quantity is explicitly stated in the measure_with_unit related to the quantified_assembly_component_usage.

NOTE - Since these constructs are sub-types of assembly_component_usage, they will require a security_classification.

Quantity designations are used on parts lists for products. The AP 210 data structure is quite capable of providing the data for the body of a parts list. The information for each record in this list is generated for an assembly by obtaining the data for the products related to it through next_assembly_usage_occurrences.

2.3.10.4 Specified Higher Usage Occurrence

There is no standard mapping for **specified_higher_order_usage_occurrence**. This entity type may be used to capture properties related to an instance within a multi-level assembly where the instance property is spacially and environmentally dependent (e.g.. temperature of a component where the assembly the component is part of is used multiple times in higher level assemblies, as in an equipment rack with multiple line cards and the temperature of the voltage regulator in line card three is of

interest.). The **specified_higher_usage_occurrence** would also allow direct identification of the reference designation as specified in IEC 60750:1983. Note that as described above, the **reference_designator** attribute of **assembly_component_usage** as defined in ISO 10303-41 (1994) is considered a partial reference designation in the context of IEC 60750. For further details of application of **specified_higher_usage_occurrence**, reference AP 209 recommended practices document.

2.3.10.5 Make from Part

For a make from part, the same rationale is applied to the design_make_from_relationship with the resultant part from the make from also being called out. Reference the Make from relationships discussion in this document.

2.3.10.6 Material Callout

For a material callout, the parts list is determined from the material_specifications related to its product_definition unless the bulk material is assigned a part number internally by the organization or a quantity unit of measure other than a simple count is needed. If a bulk material is assigned an internal part number by an organization or a unit of measure other than a simple count is needed, the usage of the material becomes a **next_assembly_usage_occurrence**. Material callout for bulk material is formalized in AP 210 with the ARM Application object of Assembly_material_composition_relationship and the relevant mapping table.

2.3.10.7 End Item Quantity

End Item Quantity is the total quantity of a component in either the entire delivered unit or some major subsection of a delivered unit. This quantity is designated in AP 210 by establishing a complex instance of promissory_usage_occurrence and quantified_assembly_component_usage. The quantity in the measure_with_unit related to the quantified_assembly_component_usage is the quantity of the part in the final article. This relationship is described pictorially in Figure 8.

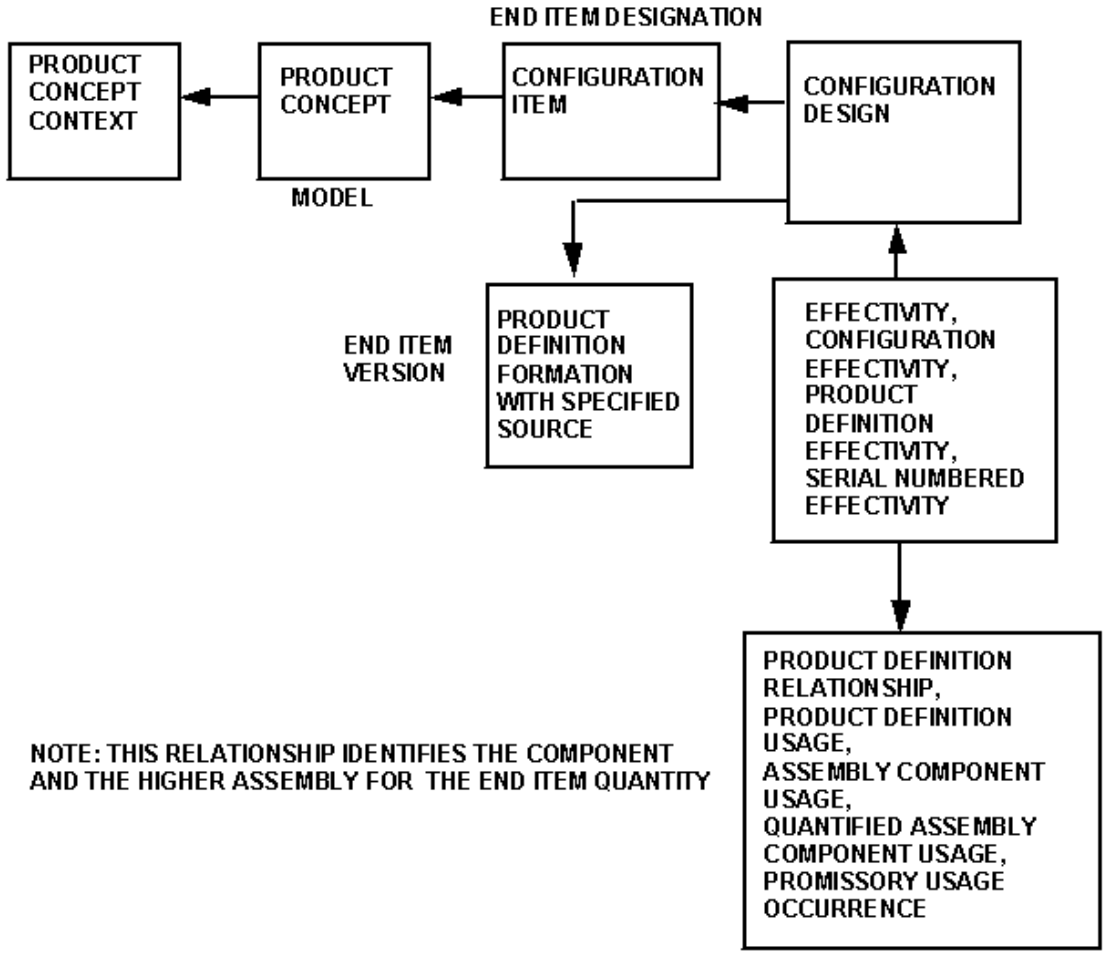


FIGURE 8 End item quantity

It should be noted that this could be a simple direct relationship or a more complex relationship. In the simple instance, the relating_product_definition will point to the product_definition of the product which is designated as the end item. In this case, the quantity is the total for the component (specified by the related_product_definition) in the end item for the indicated effectivity. In the more complex instance, the relating_product_definition will point a product_definition of a higher assembly which is not the end item. In this case, the quantity is for the component (specified by the

related_product_definition) in the assembly (specified by the relating_product_definition) as the assembly is used in the end item for the indicated effectivity.

2.3.10.8 Promissory_usage_occurrence

Since promissory_usage_occurrence is a sub-type of assembly_component_usage, it will require a security_classification. This relationship is a type of product_definition_relationship and as such may have specifications related to it.

NOTE - The relationship described here is intended for part based systems and has been show to be problematic for drawing based systems.

2.3.11 Substituting Parts in Assemblies

Refer to the following subclause in this document: "Alternate product concepts in AP 210.

2.3.12 Assemblies and Shape

The shape of an assembly is most often derived from the shape of its components. AP 210 provides three methods for dealing with the shape of an assembly. These are: replicating the shape of the components in the shape of the assembly, mapping the shape of the component into the shape of the assembly, and referencing the shape of a component. The following subsections will deal with each of these methods. AP 210 provides component_definition to represent the occurrence of a part in the the assembly. There is the possibility for the part shape, the **assembly_component** shape and the assembly shape to exist at the same time. Details of implementation scenario will determine the usefulness of maintaining a shape occurrence for the component_definition separately from the shape of the part.

2.3.12.1 Replicated Shape

One method for representing the shape of an assembly is to collect together all the elements of all the shapes of all the components explicitly in the shape of the assembly. This is the typical practice used in industry today, but it is highly inefficient. Using this method, the shape entities of the components of an assembly become shape entities in the assembly. The assembly shape becomes a conglomerate with no segregations of the various component shapes. This is represented by collecting all the geometry and topology of the components in an appropriate sub-type of shape_representation. Use of this capability is deprecated, but is provided in order to be backward compatible with AP 203.

2.3.12.2 Mapped Shape

In this method, the shape of the component is mapped into the shape of the assembly. This is done through the use of the **mapped_item** entity which can be used to map one **shape_representation** into another. (See G.1.15 on page 84 for an implementors agreement that affects this entity.) This method is the most efficient and versatile, but it may only be used where the component and the assembly shapes are of the same type of **shape_representation**. In this case, there need not be a shape occurrence for the **component_definition** that is a separate shape occurrence from that for the part. AP 210 provides a label for **shape_representation_relationship** of 'component part planar shape' to indicate that this shape is to be used for both **component_definition** and the part definition.

The **mapped_item** entity is used for transformation without scaling on the component shape. Transformation without scaling is accomplished by relating an axis2_placement in the component identified as the mapping_origin in the representation_map to an axis2_placement in the assembly identified as the mapping_target in the mapped_item. AP 210 extends this by using a **cartesian_transformation_operator_2d** (in place of the **axis_placement**) for the planar case.

2.3.12.3 Referenced Component Shape

In this method, the shape of the part is referenced to the shape of the **component_definition** through the same shape_representation_relationship, but there are two separate shape occurrences. The shape of the component_definition is still mapped using mapped_item to the shape of the assembly, so those two shapes have to be compatible.

2.3.12.4 Referenced shape

In this method, the shape of the part is referenced or related to the shape of the assembly. This is done through the use of the context_dependent_shape_representation entity. This method is not dependent on both the component and assembly shapes being the same type of shape_representation.

In this method, a complex instance of representation_relationship, representation_relationship_with_transformation, and shape_representation_relationship relates the shape_representations for the component and assembly together and relates an item_defined_transformation to the relationship. The context_dependent_shape_representation entity relates the complex relationship with the transformation to a product_definition_shape which is related to the next_assembly_usage_occurrence. Reference the AP 203 recommended practices for more details on this approach. Since the primary vehicle in AP 210 is use of library data, it is not expected that this option will be of much interest for purely AP 210 applications, but will primarily be of interest for data sharing between electrical and mechanical systems in order to preserve existing geometric models.

2.3.13 Assemblies and Specifications

AP 210 provides explicit ARM concepts to support product specifications. Reference the ARM concepts of Requirement_allocation and Specification_allocation for further information. There are standard mappings provided in clause 5.1 of the standard to support these requirements.

2.3.14 Engineering Release/ Change Data

AP 210 provides data structures for representation of the data used in the engineering release and change process. The structures are based on a request and action process where a request is established documenting the need for a potential release or change which may or may not ever be incorporated. If the request is incorporated, it is done through some action being taken on the request which results in either a new release of a design or a change to a existing design.

It should be noted that these constructs have been designed to represent all request and incorporation structures in the AP 210 application domain. All release and change proposals and requests (Engineering Change Proposals, Requests for Engineering Action, etc.) are represented by the request portion of the structure. All release and change incorporations are represented by the action portion of the structure. Differentiation between types of requests and actions can be done structurally based on the guidance in this section, by its identification (id for requests, name for actions), or by the originator. Differentiation by identification or originator is very process dependent but can be necessary particularly for preliminary requests and proposals.

Some types of releases and changes in organizations may not involve a two step process. In this case, both data structures are implemented simultaneously and reference the same release or change documentation. Since these constructs in AP 210 are intended to support many different release and change processes/documentation, in some cases, some of the required data may not exist.

The release process is initiated through an AP 210 versioned_action_request which is related to the design being released through a start_request. The versioned_action_request has a related action_method. In this case, both the versioned_action_request and the action_method would indicate that the respective purposes were to initially release the design or create the design for the initial release. This request process is followed (in the data) by an action_directive which is related to the design to be released through a start_work. The action_directive also identifies the start_request as the request being satisfied/ incorporated. A directed_action relates the action_method to the

action_directive which in the case of initial release may be moot. The structure of these relationships (at a high level) is shown in Figure 9.

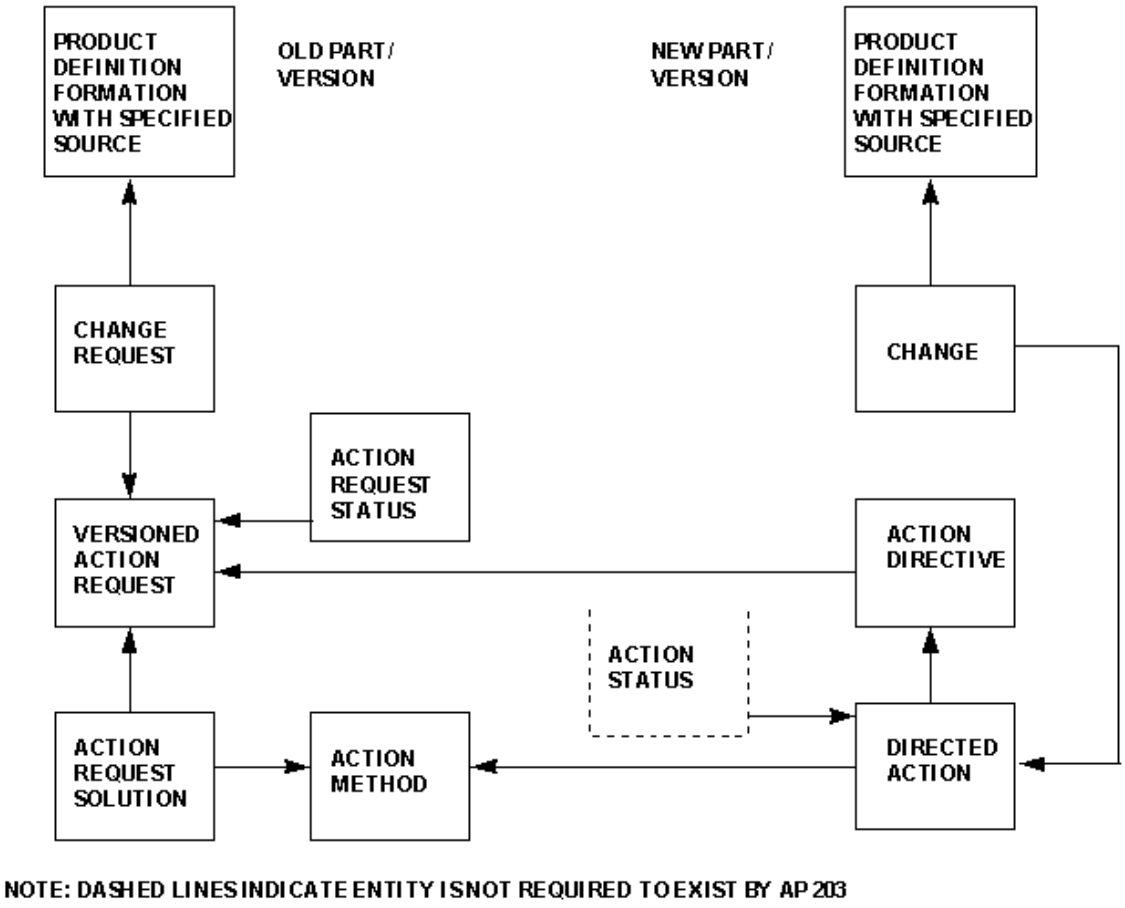


FIGURE 9 Change requests

The change process is initiated through an AP 210 versioned_action_request, as well, which is related to the design proposed to be changed through a change_request. The versioned_action_request has a related action_method. In this case, there may be many action_methods or ways to solve the problem. This request process is followed (in the data) by an action_directive which is related to the new design or version to be released through a change. The action_directive also identifies the change_request(s) as the request(s) being satisfied/ incorporated. A directed_action relates the action_method to the action_directive indicating which of possibly many methods for the request or requests incorporated was chosen. The structure of these relationships (at a high level) is shown in Figure 9.

2.3.14.1 Requests for Release/Change

Requests for release or change are created in AP 210 by relating a versioned_action_request to a product_definition_formation through a start_request or change_request. The start_request or change_request identifies through the items attribute the product_definition_formation to be released

or changed. In the case of a start_request, this is a bit odd as this structure in AP 210 requires an identification of a version at request time which will in fact result from the request. This may be changed in future editions of the AP.

The versioned_action_request id attribute contains the identification of the request. This information is the document or request number. The version attribute is the version of the request itself. This attribute is used to identify actual versioning of the request or reissues of the request. The purpose attribute should contain text identifying the end result anticipated from this version of this request. The description attribute should contain a general description of the request. In AP 210, a versioned_action_request is required to have an associated action_request_status. The AP restricts the values for the status attribute to "proposed", "in_work", "issued", or "hold".

A request for release or change may have many possible ways it can be resolved. This is more common for changes than releases, but the AP 210 data structure supports the documentation of the engineering thought process gone through in either case. This is accomplished through a combination of the action_request_solution and action_method entities. Action_request_solution relates an action_method to a versioned_action_request. The action_method name attribute should contain a reference to any formal documentation for a proposed solution to the release/ change request. The description attribute should contain a detailed description of the method through which the request is to be satisfied. The consequence attribute should contain any determined or perceived consequence to using this method to satisfy this request. The purpose attribute should contain the intention of the method as a single method may be used to satisfy many requests.

AP 210 requires that a start_request and a change_request have a related approval. As these requests normally have a number of signatories, there should be no problem obtaining this data if it is stored in electronic form. A start_request or change_request is required to be associated with a date and time in the role of "request_date" which indicates when the request was created. Lastly, a start_request or change_request is required to be associated with at least one person and organization in the role of "initiator" or "request_recipient".

AP 210 provides the ability to identify specifically what product data entities differ from one revision to the next, with the use of:

- **add_design_object_assignment,**
- **add_design_object_request_assignment,**
- **change_from_design_object_assignment,**
- **change_to_design_object_assignment,**
- **change_from_design_object_request_assignment,**
- **change_to_design_object_request_assignment.**
- **delete_design_object_assignment,**
- **delete_design_object_request_assignment,**
- **product_definition_relationship name** attribute of {design object addition, design object deletion, design object change},

AP 210 provides the ability to state the characteristics that are the reason for the change or work item with the **representation_relationship** with **name** attribute of 'evaluated characteristic'. The **representation_relationship** is one of the **items** in the **start_request** or **change_request** if it is supplied. Otherwise the reason is encoded in the **purpose** attribute of **versioned_action_request**. The DIS version of AP 210 AIM did not contain **representation_relationship** in the **items** attribute of these entities. This has been added for the IS version.

2.3.15 Release/Change Incorporation

Release of a design or change incorporation into a design is accomplished in AP 210 through the start_work and change constructs which relate an action_directive to the new design or new design version by pointing to the product_definition_formation which results from the release or change. A directed_action related to the action_directive identifies the action_method actually used to satisfy the requests related to the action_directive. In the case where many requests are being incorporated, there many be many directed_actions to indicate the appropriate methods.

The action_directive name attribute is the identification of the formal documentation to incorporate the change or release the design. In cases where there is no second set of paper work or documentation (i.e. there is a one to one correspondence between versioned_action_request and action_directive), the action_directive name value is the same as the versioned_action_request id value. The description attribute should contain a phrase or group of phrases indicating the final result of the release or change. The analysis attribute should identify any investigative results which support the release or change. Likewise, the comment attribute should contain any textual commentary which supports the release or change. An action_directive may be associated with an action_status which serves the same function as action_request_status in the previous request section. It is not required, in AP 210, that the action_directive be related to an action_status as the two sets of data may represent one or two documents.

The directed_action name attribute should contain the identification of the formal documentation as to why the method identified was chosen. The description attribute should contain a textual description, either in summary or detail, supporting the chosen method.

In AP 210, a start_work or change is required to have an associated approval. If these constructs are representing the same document, they could share the approval. A start_work or change is required to have a date and time associated with it in the role of "start_date" which is when the work to satisfy the request or requests began. Once completed, a start_work or change may have a date and time associated with it in the role of "release_date".

2.3.16 Release/Change Reissues

Engineering release and changes are reissued. This can be caused by an error or omission in the change package. It may also be done to signify changes in effectivity which have no effect on the version of the part.

AP 210 supports the reissue of releases and changes. To reissue a release or a change, a **versioned_action_request** is created with an **id** attribute value equal to the **action_directive** name being reissued. The **versioned_action_request version** attribute contains the reissue identifier. This new **versioned_action_request** is added to the set of requests in the original **action_directive** that was issued.

2.3.17 Configuration Identification

Configuration identification in AP 210 is done through the **configuration_item** entity. This entity identifies products as end items or items which are sold or delivered. As in industry, this designation can be applied to full systems or spares (which are also referred to as the lowest level replaceable units).

The **configuration_item id** attribute is a unique identification of the item which may be a part number but more probably a moniker. The **name** attribute is a short description of the item. The **description** attribute is optional and would be the expanded name or description of the item. The **purpose** attribute is also optional and would contain a description of the item's intended use.

A **configuration_item** is related to a **product_concept**. The **product_concept id** attribute is more commonly known as the model designation. The **product_concept** taken together with the **configuration_item** describe a model series or configured production run. The **name** attribute is a short description of the model. The **description** attribute is the expanded name or description of the model. The **product_concept** is related to a **product_concept_context** where the **market_segment_type** attribute identifies what customer or group of customers provided the requirements for the model.

AP 210 requires that a **configuration_item** have an approval. A **configuration_item** must also be associated with a **person_and_organization** in the role of "configuration manager".

Pre-processor Recommendations: In some cases, it may be difficult to determine the approval and "configuration manager" for a **configuration_item**. If the item has effectivity (see next section), this information may be extrapolated from the approval and "creator" information for the **product_definition_formation** for that product. If not, this information should be obtained from user input or a default based on the **configuration_item id** attribute.

2.3.18 Effectivity

Effectivity is the designation that something or a relationship between two things is used or planned to be used in some **configuration_item**. In AP 210, effectivity is designated on relationships between **product_definitions** by either ranges of serial numbers, ranges of dates or a lot. This is accomplished through a complex instance of the entities **effectivity, configuration_effectivity, product_definition_effectivity** and one of either **serial_numbered_effectivity, dated_effectivity** or **lot_effectivity**.

A **serial_numbered_effectivity** specifies an effectivity_start_id with an optional effectivity_end_id. If the effectivity_end_id does not exist, the effectivity is good for the starting serial number and all following serial numbers. A dated_effectivity follows the same pattern using dates rather than serial numbers. A lot_effectivity indicates an effectivity_lot_id and an effectivity_lot_size.

The above entities specify the effectivity identifiers. These entities are related to a product_definition_relationship through the usage attribute in the product_definition_effectivity entity. The effectivity entity id attribute has no standard mapping. The configuration_effectivity entity relates these relationships to a configuration_design which relates a configuration_item to a

product_definition_formation or design version. Figure 10 shows this relationship pictorially for a serial_numbered_effectivity.

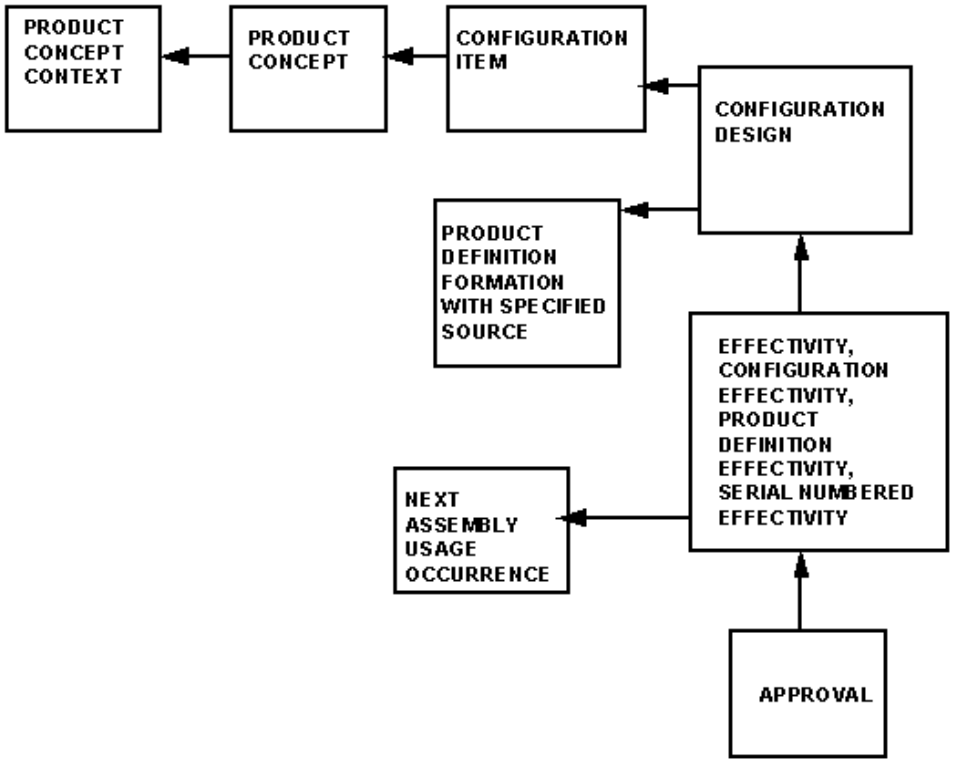


FIGURE 10 Effectivity

The whole relationship here can be simply stated as a range of serial numbers, dates or a lot number related to a product_definition_formationor design version which is designated as a configuration_item. This does mean that all configuration_items must be associated to a design version in order to have effectivity. In AP 210, an effectivity requires an approval. This information may be difficult to find in some instances (see pre-processor recommendations for guidance.

It should be noted that since the effectivity is related to a product_definition_relationship, many different views of the effectivity can be established by varying the relating_product_definition. AP 210 can maintain one effectivity based on the "design" product_definition and another based on the "manufacturing" product_definition. The various product_definitions can move into other life cycle stages for the design as well. In this way, effectivities can be defined for any of a number of views and life cycle stages of the design.

The conformance classes in AP 210 do not require that effectivity relationships be instantiated. The reason for this is that there are occasions where data needs to be exchanged or shared prior to an effectivity being defined. This tends to occur early in a new design.

All effectivities in AP 210 are explicit effectivities and there are no assumed effectivities. Some systems in existence today assume a part is effective for all planned or actual instances of a product model if the effectivity is not explicitly defined. This is not the intent in AP 210. If a part has no effectivity in the AP 210 data structures, it has no effectivity. If a part is effective for all instances of a product model, the data should explicitly state all the effective instances. The effectivities in AP 210 contain open ranges for serial numbers and dates to allow for open or full effectivities. Using these constructs, all that is required is a start point. If there is a desire for full effectivity and the start point is not defined, the value "1" should be used for the serial_numbered_effectivity effectivity_start_id or the equivalent date of January 1st year 1 should used for dated_effectivity effectivity_start_date.

NOTE - Open effectivity does not make sense for a lot effectivity as it is inherently closed (other than lot size). Lot effectivity is typically an effectivity designated in the manufacturing view of a product or part.

The exchange or sharing of effectivity information creates the need for optional processing capability in at least pre-processors to allow for perspective. It is typically desirable for the lead contractor in a partnership or team to provide effectivity definitions to sub-contractors. It is usually undesirable for the lead contractor to utilize effectivities echoed back by sub-contractors as they reflect what was originally sent but not necessarily the most current data (in some cases).

The above is a simple case. Most cases involve even more variables such as who in the exchange or sharing arrangement is the defining body for the effectivity of a particular part or usage. One way to deal with this situation is for pre-processors to provide options for ignoring effectivity entirely, loading it or either ignoring or loading it based on externally defined criteria such as the part's design owner, design supplier or part number and for post-processors to provide a switch for a user choice on whether or not defined effectivity information in the system should be used in the interchange.

Pre-processor Recommendations: There is no standard mapping for the id attribute of the effectivity entity. Since there is no standard mapping in the AP 210 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate/ mutually agreed upon string. If the effectivity approval information is not readily available, it can be extrapolated from the engineering change which designated the effectivity. Pre-processors should interpret the value "1" for the serial_numbered_effectivity effectivity_start_id or the equivalent date of January 1st year 1 for dated_effectivity effectivity_start_date as full or open effectivity when the values are specified with no ending range value. It is recommended that pre-processors provide options for ignoring effectivity entirely, loading it, or either ignoring or loading it based on externally defined criteria such as the part's design owner or part number to allow for a user choice as to whether the data is utilized or not depending on the source.

Post-processor Recommendations: There is no standard mapping for the id attribute of the effectivity entity. Since there is no standard mapping in the AP 210 application domain for this attribute, it is recommended that post-processors assign no processing significance to this value. When there is a need for full effectivity and the start point is not defined, post-processors should use the value "1" for the serial_numbered_effectivity effectivity_start_id or the equivalent date of January 1st year 1 for dated_effectivity effectivity_start_date. It is recommended that post-processors provide a switch for a user choice on whether or not defined effectivity information in the system should be used in the interchange.

2.4 Detailed Assembly Example

2.4.1 Printed Circuit Assembly

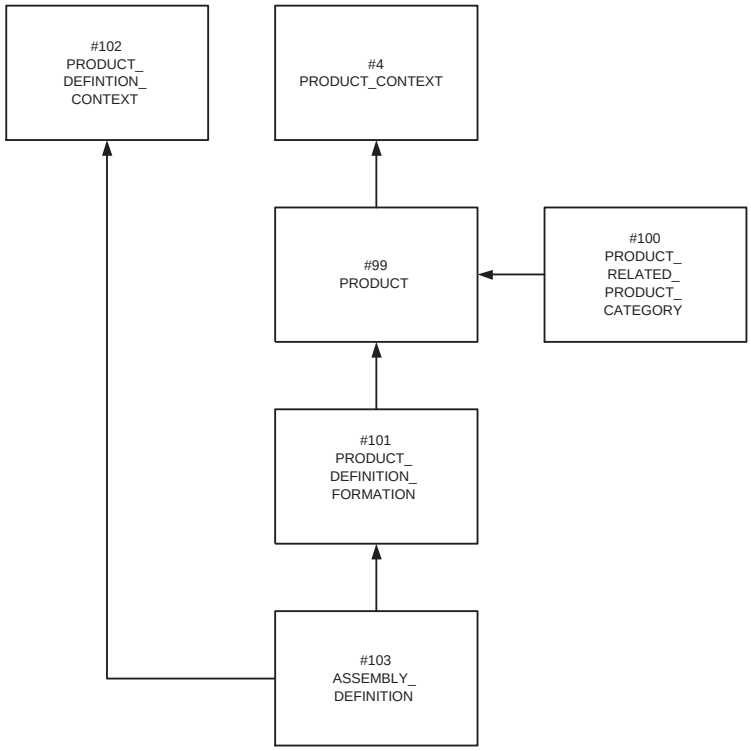


FIGURE 11 **Assembly Definition Product**

- **Sample Part 21 Code**
  - **Assembly Definition Product Entities**

```
#100=PRODUCT_RELATED_PRODUCT_CATEGORY('assembly module','pca',(#99));
#99=PRODUCT('','','',(#4));
#4=PRODUCT_CONTEXT('',#2,'electromechanical');
#101=PRODUCT_DEFINITION_FORMATION('','',#99);
#103=ASSEMBLY_DEFINITION('','',#101,#102);
#102=PRODUCT_DEFINITION_CONTEXT('physical design',#2,'design');
```

A printed circuit assembly is nearly a synonym for printed wiring assembly, except that a printed circuit assembly may include printed components in its printed circuit board and a printed wiring assembly is constructed upon a printed wiring board that is totally interconnect without printed

components. Another synonym for printed circuit assembly is circuit card assembly. For the purposes of AP210 all three have been mapped to the ARM concept of PCA and AIM assembly_definition entity. This recommended practice is limited to describing planar printed assemblies.

2.4.1.1 The product entity

AP210 identifies a PCA AO with a **product** entity that is referenced by the products attribute of a **product_related_product_category** entity with a name of "assembly module" and by the of_product attribute of a **product_definition_formation** which is in turn referenced by the formation attribute of **assembly_definition** entities.

2.4.1.2 The product context entity

The products frame_of_reference attribute references a **product_context** entity.

2.4.1.3 The product related product category

The product is referenced by the products attribute of a **product_related_product_category** entity with a name of "assembly module".

2.4.2 The Printed Circuit Assembly Physical Design Usage View

[to be added later.]

2.4.3 The Printed Circuit Assembly Physical Design View

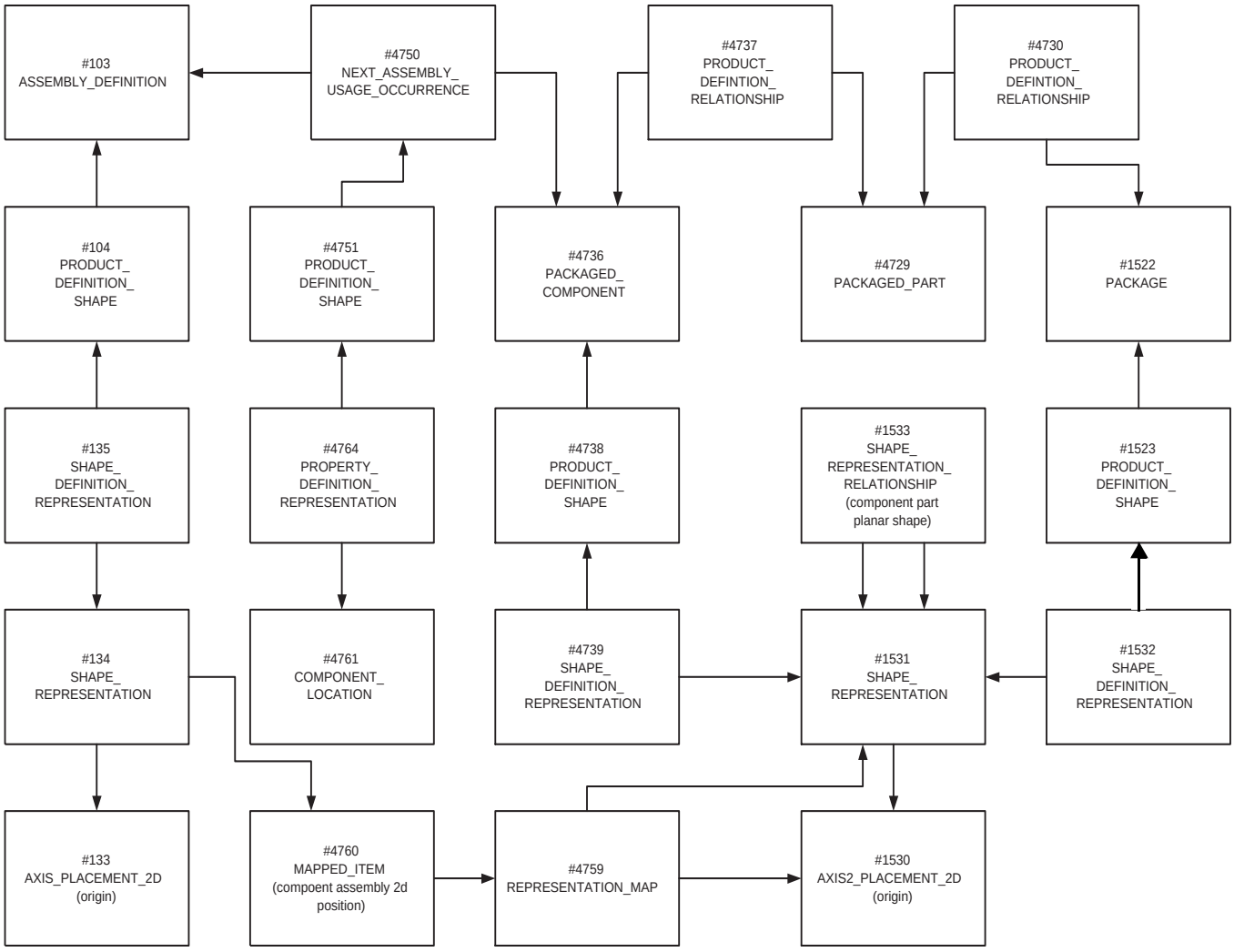


FIGURE 12 PCA Physical Design View

- **Sample Part 21 Code**
  - **Physical Design view of PCA**

```
#103=ASSEMBLY_DEFINITION('','',#101,#102);
#104=PRODUCT_DEFINITION_SHAPE('','',#103);
#135=SHAPE_DEFINITION_REPRESENTATION(#104,#134);
#134=SHAPE_REPRESENTATION('',(#133,#1340,#4760,#4803,#4847,#4883,#4926,#4962,
   #4998,#5041,#5077,#5113,#5156,#5199,#5242,#5278,#5314,#5350,#5386,#5422,
   #5458,#5494,#5537,#5573,#5617,#5653,#5696,#5732,#5768,#5804,#5840,#5884,
   #5920,#5964,#6008,#6051,#6094,#6138,#6174,#6217,#6253,#6289,#6332,#6368,
   #6411,#6454,#6497,#6533,#6576,#6619,#6655,#6698,#6741,#6784,#6827,#6870,
   #6913,#6956,#6999,#7035,#7078,#7114,#7150,#7193,#7237,#7273,#7318,#7361,
   #7397,#7433,#7469,#7512,#7548),#130);
#133=AXIS2_PLACEMENT_2D('origin',#131,#132);
```

  - **Packaged component**

```
#4750=NEXT_ASSEMBLY_USAGE_OCCURRENCE('','','',#103,#4736,'C1');
#4736=PACKAGED_COMPONENT('C1','',#4734,#4735);
#4738=PRODUCT_DEFINITION_SHAPE('','',#4736);
#4739=SHAPE_DEFINITION_REPRESENTATION(#4738,#1531);
```

  - **Component Location**

```
#4751=PRODUCT_DEFINITION_SHAPE('','',#4750);
#4764=PROPERTY_DEFINITION_REPRESENTATION(#4751,#4761);
#4761=COMPONENT_LOCATION('',(#4753,#4754,#4758,#4760),#4752);
#4760=MAPPED_ITEM('component assembly 2d position',#4759,#4758);
#4759=REPRESENTATION_MAP(#1530,#1531);
#1530=AXIS2_PLACEMENT_2D('origin',#1528,#1529);
```

  - **Packaged component Product**

[to be added later]

  - **Packaged Part**

```
#4737=PRODUCT_DEFINITION_RELATIONSHIP('','instantiated part','',#4729,#4736);
#4729=PACKAGED_PART('cap-1','',#4727,#4728);
```

  - **Package**

```
#4730=PRODUCT_DEFINITION_RELATIONSHIP('','used package','',#1522,#4729);
#1522=PACKAGE('','',#1520,#1521);
#1523=PRODUCT_DEFINITION_SHAPE('cap-1','ComponentGeometry.Graphics',#1522);
#1532=SHAPE_DEFINITION_REPRESENTATION(#1523,#1531);
```

  - **Package Geometry**

```
#1533=SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape','',#1531,
   #1531);
#1531=SHAPE_REPRESENTATION('feature shape',(#1530,#1582,#1541,#1549,#1556),
   #1527);
```

2.4.4 Component Location

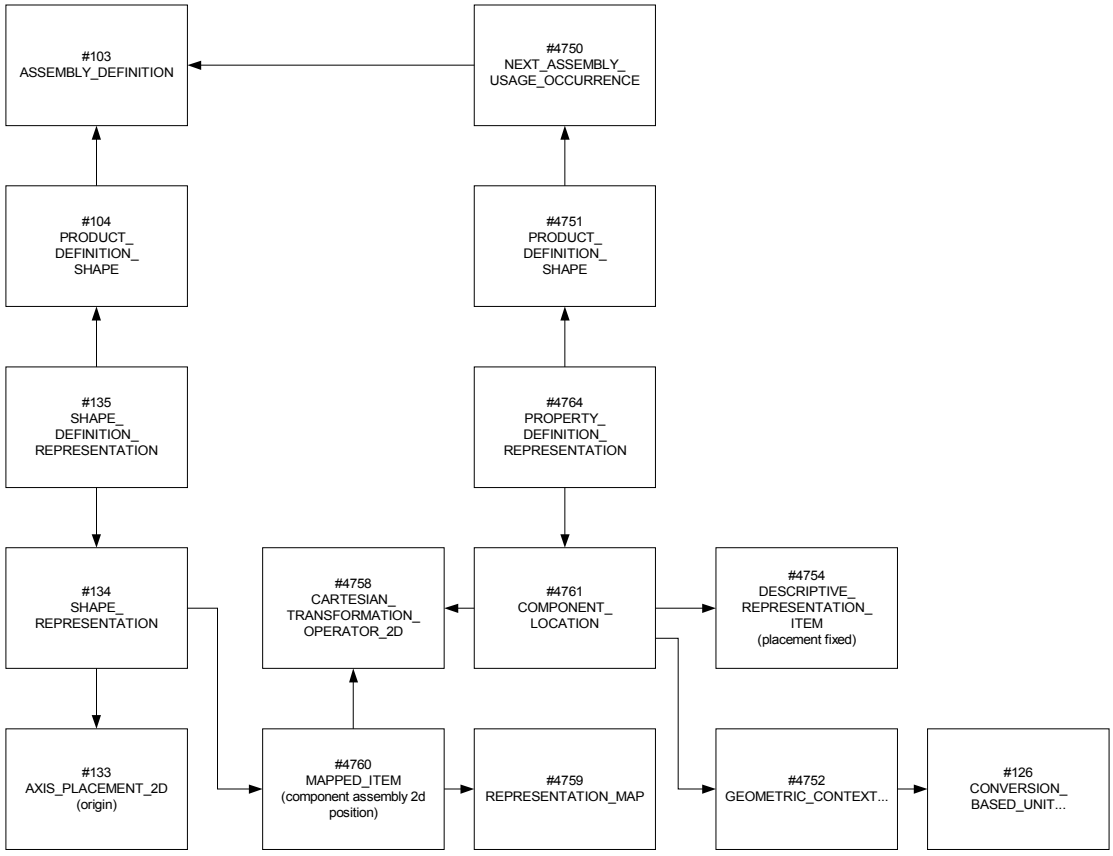


FIGURE 13 **Component_2d_location and Component_assembly_2d_position**

- **Sample Part 21 Code**
  - **Physical Design view of PCA**

```
#103=PHYSICAL_UNIT('','',#101,#102);
#104=PRODUCT_DEFINITION_SHAPE('','',#103);
#135=SHAPE_DEFINITION_REPRESENTATION(#104,#134);
#134=SHAPE_REPRESENTATION('',(#133,#1340,#4760,#4803,#4847,#4883,#4926,#4962,
   #4998,#5041,#5077,#5113,#5156,#5199,#5242,#5278,#5314,#5350,#5386,#5422,
   #5458,#5494,#5537,#5573,#5617,#5653,#5696,#5732,#5768,#5804,#5840,#5884,
   #5920,#5964,#6008,#6051,#6094,#6138,#6174,#6217,#6253,#6289,#6332,#6368,
   #6411,#6454,#6497,#6533,#6576,#6619,#6655,#6698,#6741,#6784,#6827,#6870,
   #6913,#6956,#6999,#7035,#7078,#7114,#7150,#7193,#7237,#7273,#7318,#7361,
   #7397,#7433,#7469,#7512,#7548),#130);
#133=AXIS2_PLACEMENT_2D('origin',#131,#132);
```

  - **Component Location**

```
#4750=NEXT_ASSEMBLY_USAGE_OCCURRENCE('','','',#103,#4736,'C1');
#4751=PRODUCT_DEFINITION_SHAPE('','',#4750);
#4764=PROPERTY_DEFINITION_REPRESENTATION(#4751,#4761);
#4761=COMPONENT_LOCATION('',(#4753,#4754,#4758,#4760),#4752);
#4753=DESCRIPTIVE_REPRESENTATION_ITEM('mounting style','normal');
#4754=DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#4758=CARTESIAN_TRANSFORMATION_OPERATOR_2D('','','',#4756,#4757,#4755,1.0);
#4760=MAPPED_ITEM('component assembly 2d position',#4759,#133);
#4759=REPRESENTATION_MAP(#1530,#1531);
#4752=(GEOMETRIC_REPRESENTATION_CONTEXT(2)
   GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#128))GLOBAL_UNIT_ASSIGNED_CONTEXT(
   (#116,#126))REPRESENTATION_CONTEXT('','2D'));
#126=(CONVERSION_BASED_UNIT('INCH',#122)LENGTH_UNIT()NAMED_UNIT(#121));
```

  - **Component Location side**

```
#4763=PROPERTY_DEFINITION_REPRESENTATION(#4762,#4761);
#4762=PROPERTY_DEFINITION('','',#4705);
#4705=COMPONENT_SHAPE_ASPECT('','interconnect module component surface feature',
   #141,.F.);
#4706=SHAPE_ASPECT_RELATIONSHIP('instantiated feature','',#96,#4705);
#96=SHAPE_ASPECT('','interconnect module primary surface',#95,.U.);
```

The location of a component is defined in AP210 by the Component_2d_location and Component_assembly_2d_position application objects. The Component_2d_location is mapped to a **component_location** entity with the transformation specifying the location mapped to a **cartesian_transformation_operator_2d** in the component_location's items attribute. The component_location's context_of_items attribute is a complex instance that specifies the units of the cartesian_transformation_operator_2d, this allows component locations to be specified in a different unit system than the assemblies geometry. The Component_assembly_2d_position is mapped to a **mapped_item** entity with a description attribute of "component assembly 2d position" with the component_location's cartesian_transformation_operator_2d as the mapping_target.

2.5 Detailed Interconnect Example

2.5.1 Printed Circuit Board

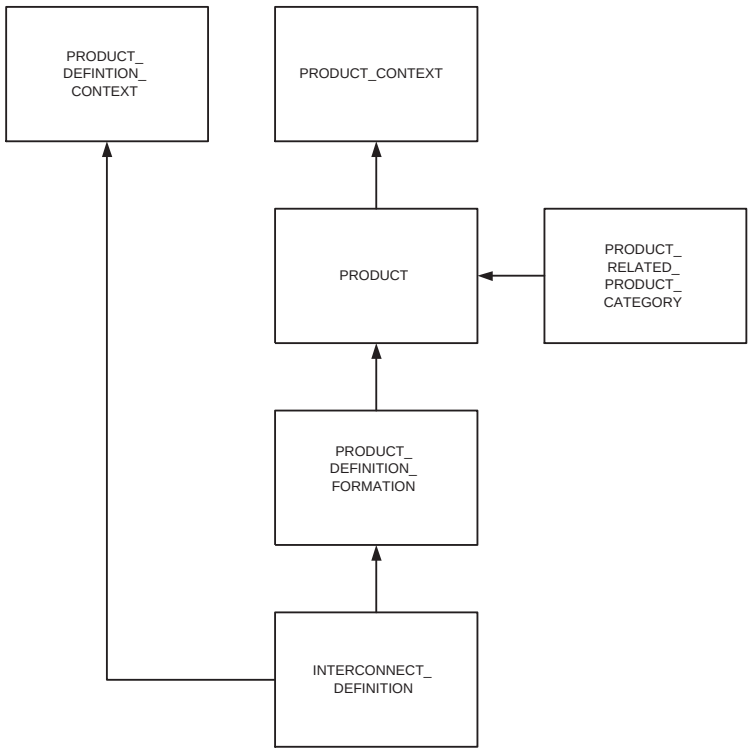


FIGURE 14 **Interconnect Module Product and Definition.**

- **Sample Part 21 Code**
  - **Interconnect Module product**

```
#12=PRODUCT_CONTEXT('',#10,'mechanical');
#77=PRODUCT('test_design_interconnect','','',(#12));
#84=PRODUCT_RELATED_PRODUCT_CATEGORY('interconnect module','pcb',(#77));
#80=PRODUCT_DEFINITION_FORMATION('1','',#77);
#83=INTERCONNECT_MODULE('','interconnect module',#80,#21);
#21=PRODUCT_DEFINITION_CONTEXT('physical design',#10,'as designed');
```

A printed circuit board is nearly a synonym for printed wiring board, except that a printed circuit board may include printed components and a printed wiring board is totally interconnected without printed components. For the purposes of AP210 both have been mapped to the ARM concept of PCB and the AIM interconnect_definition entity. This recommended practice is limited to describing planar printed circuit boards and printed wiring boards.

In order to define an interconnect module for a PCB AO in AP210, five AIM entities are used. The **product** entity establishes its identification (or part number), name (or nomenclature), and description. **Interconnect_definition** entities define the physical design and physical design usage models of the interconnect module.

2.5.1.1 The product entity

AP210 identifies a PCB AO with a **product** entity that is referenced by the products attribute of a **product_related_product_category** entity with a name of "interconnect module" and by the of_product attribute of a **product_definition_formation** which is in turn referenced by the formation attribute of **interconnect_definition** entities.

2.5.1.2 The product context entity

The products frame_of_reference attribute references a **product_context** entity.

2.5.1.3 The product related product category

The product is referenced by the products attribute of a **product_related_product_category** entity with a name of "interconnect module".

2.5.2 The Printed Circuit Board Physical Design Usage View

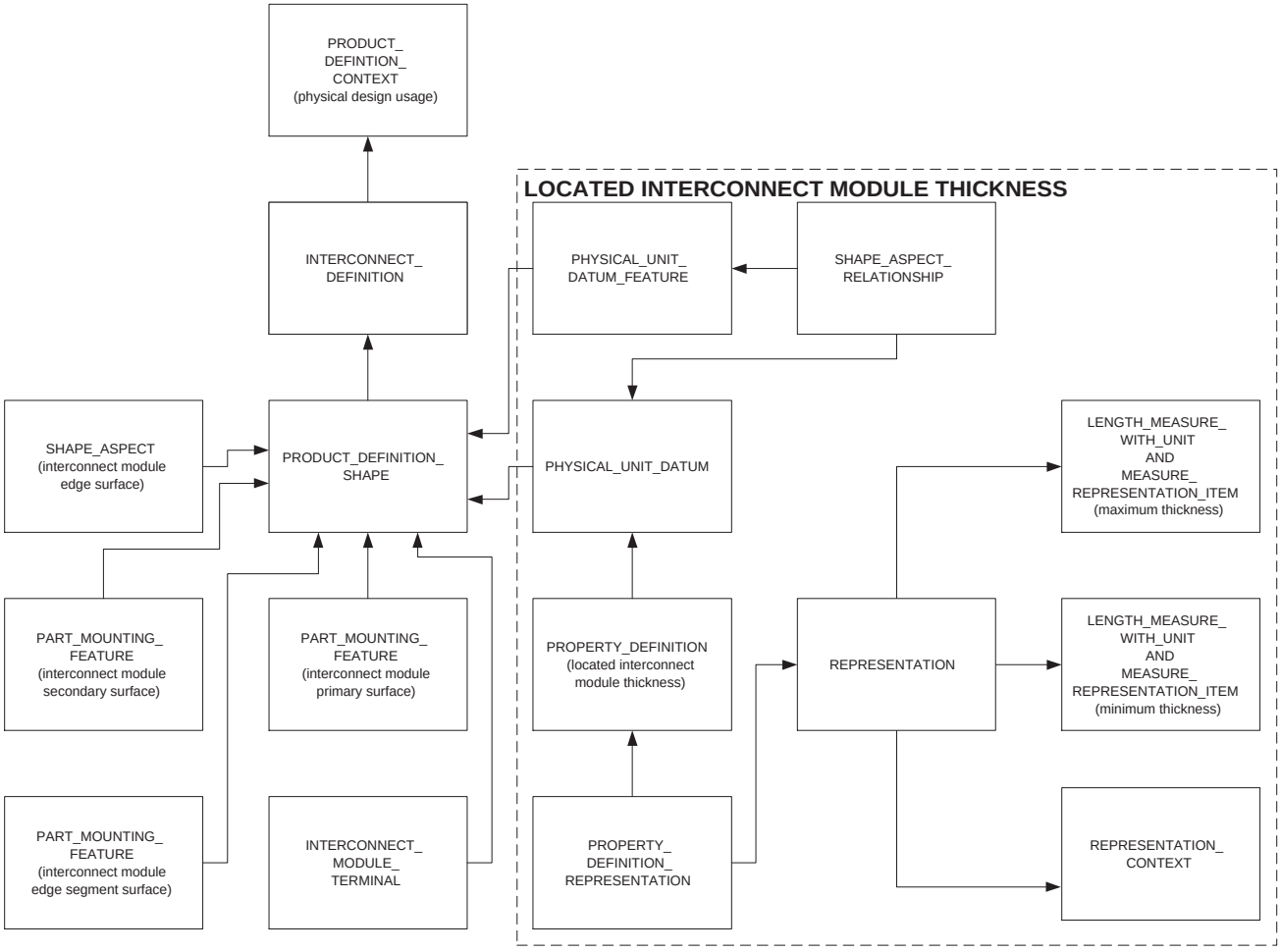


FIGURE 15 PCB Physical Design Usage View

The physical design usage view identifies a PCB AO as a part in its next assembly with sufficient detail to allow its use but without sufficient detail to permit reproduction of the printed circuit board.

AP210 defines the PCB Physical Design Usage View using three entities an interconnect_definition, shape_aspect and product_definition_shape. A **shape_aspect** entities define the interconnect modules primary, secondary and edge surfaces in the physical design usage model.

2.5.2.1 The interconnect_definition entity

The PCB physical design usage view is defined by an **interconnect_definition** entity whose frame_of_reference attribute references a **product_definition_context** entity with a name of "physical design usage".

2.5.2.2 The primary surface shape_aspect entity

AP210 specifies the primary surface feature of the PCB AO in the physical design usage view with a **shape_aspect** entity with a description of "interconnect module primary surface" that references a **product_definition_shape** entity that references the physical design usage views **interconnect_definition**.

2.5.2.3 The secondary surface shape_aspect entity

AP210 specifies the secondary surface feature of the PCB AO in the physical design usage view with a **shape_aspect** entity with a description of "interconnect module secondary surface" that references a **product_definition_shape** entity that references the physical design usage views **interconnect_definition**.

2.5.2.4 The edge surface shape_aspect entity

AP210 specifies the edge surface feature of the PCB AO in the physical design usage view with a **shape_aspect** entity with a description of "interconnect module edge surface" that references a **product_definition_shape** entity that references the physical design usage views **interconnect_definition**. The shape_aspect is referenced by the definition attribute of a **property_definition** entity. The **property_definition** is referenced by a **property_definition_representation** that references a **shape_representation.** If the shape is planar it represents a projection of the maximum extent of the interconnect module's edge onto the planar shape.

2.5.2.5 The interconnect module thickness property_definition_representation entity

AP210 specifies the thickness of the PCB AO in the physical design usage view with a **product_definition_representation** entity with a definition attribute that references a **property_definition** entity with a name of "located interconnect module thickness". The property_definition references a **physical_unit_datum** entity with an of_shape attribute that references a **physical_definition_shape** entity with a definition attribute that is the interconnect_definition. The physical_unit_datum is referenced by the rep_1 attribute of a **shape_aspect_relationship** entity with a rep_2 attribute that references a **physical_unit_datum_feature** entity with an of_shape attribute that also references the interconnect_definition's physical_definition_shape. The physical_unit_datum's of_shape attribute references the interconnect_definition. The property_definition is also referenced by the definition attribute of a **property_definition_representation** entity with a used_representation attribute that references **representation** entity with a items attribute that references two **length_measure_with_unit_and_measure_representation_item** complex entities one with a name of "maximum thickness" and one with a name of "minimum thickness" that both contain the boards thickness.

2.5.3 The Printed Circuit Board Physical Design View

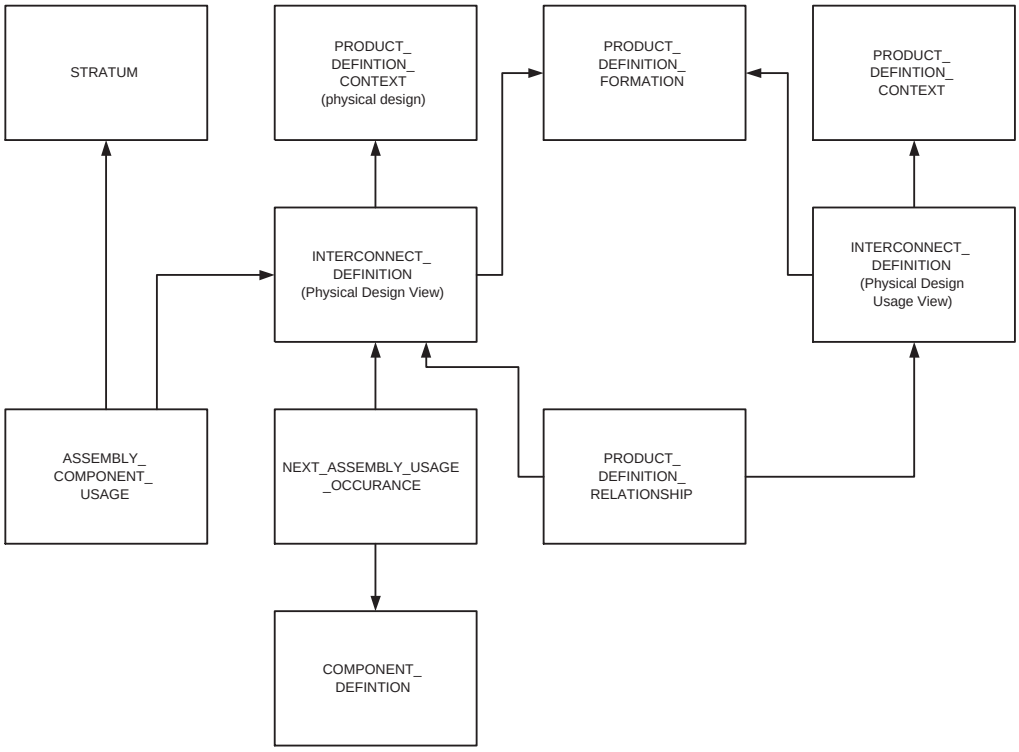


FIGURE 16 **Printed Circuit Board Physical Design**

- **Sample Part 21 Code**

```
#21=PRODUCT_DEFINITION_CONTEXT('physical design',#10,'as designed');
#83=INTERCONNECT_DEFINITION('','interconnect module',#80,#21);
#258=NEXT_ASSEMBLY_USAGE_OCCURRENCE('','assembly composition','',#83,#123,
$);
#123=COMPONENT_DEFINITION('','interconnect module edge',#80,#15);
#682=PRODUCT_DEFINITION_RELATIONSHIP('','design usage','',#83,#89);
#89=INTERCONNECT_DEFINITION('','interconnect module',#80,#88);
#88=PRODUCT_DEFINITION_CONTEXT('physical design usage',#10,'as designed
#80=PRODUCT_DEFINITION_FORMATION('1','',#77);
#22=STRATUM('top','',#80,#21);
```

The physical design view fully describes a PCB AO in sufficient detail to permit production in a manufacturing process and analysis of it's response to a simulated environment in a computer based experimental environment.

A **component_definition** entity defines the interconnect modules edge in the physical design model. **Stratum** entities define the PCB AO layers.

2.5.3.1 The interconnect_definition entity

The PCB physical design view is defined by an **interconnect_definition** entity whose frame_of_reference attribute references a **product_definition_context** entity with a name of "physical design" and assembly relationships with the printed circuit boards stratum and interconnect module edge definitions. It is related to the physical design view with by a **product_definition_relationship** entity with a name of "design usage" and has a property_definition that is the shape of the printed circuit board.

2.5.3.2 The component_definition

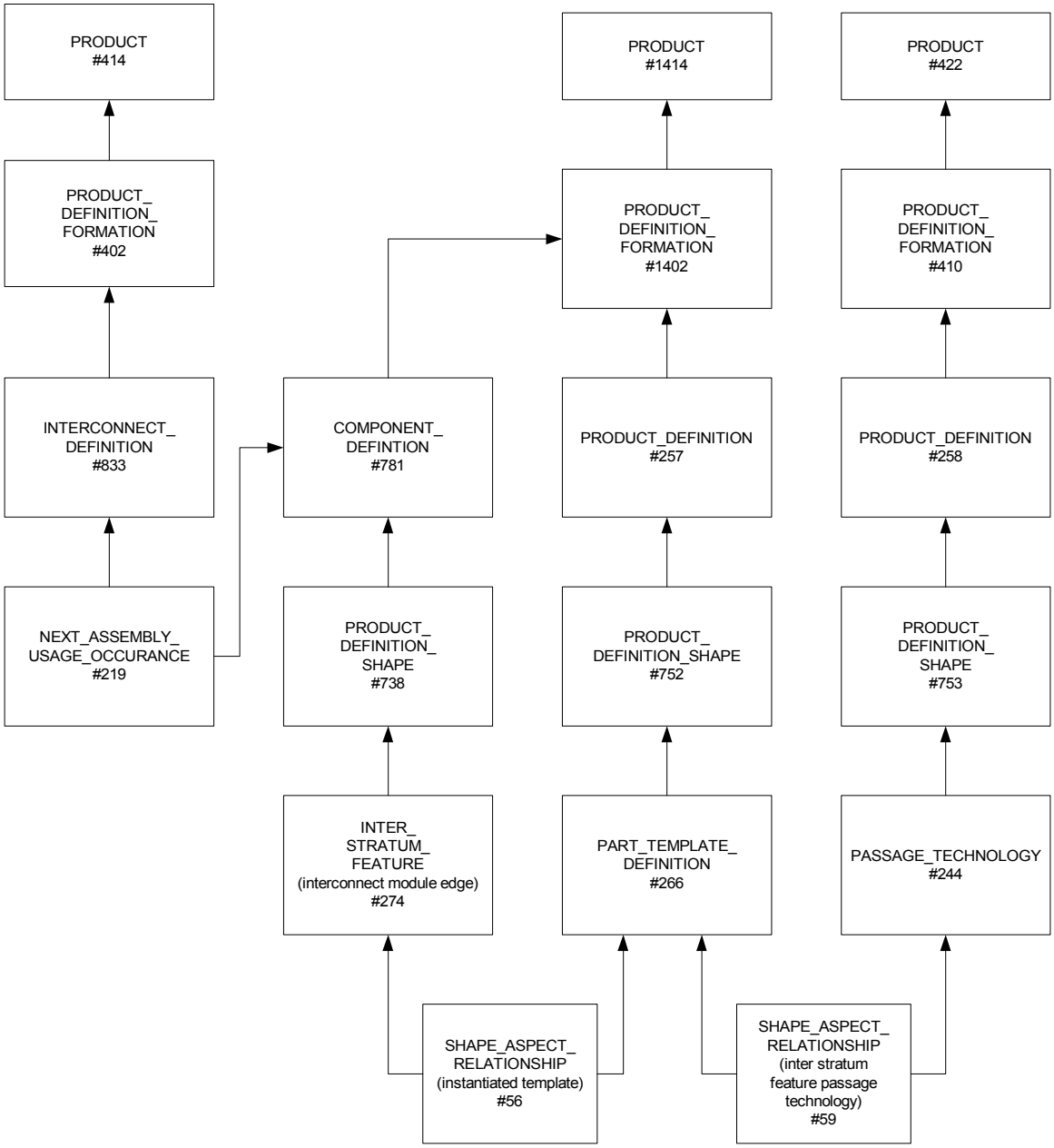


FIGURE 17 Interconnect Module Edge

AP210 specifies the edge of the PCB AO in the physical design view with a **component_definition** entity related to the physical design views **interconnect_definition** by a **next_component_usage_occurance** entity. The **component_definition** has it's own **product** entity.

The interconnect module edge is specified as a Assembly_component AO that is mapped to the **inter_stratum_feature** subtype of the **component_shape_aspect** entity.

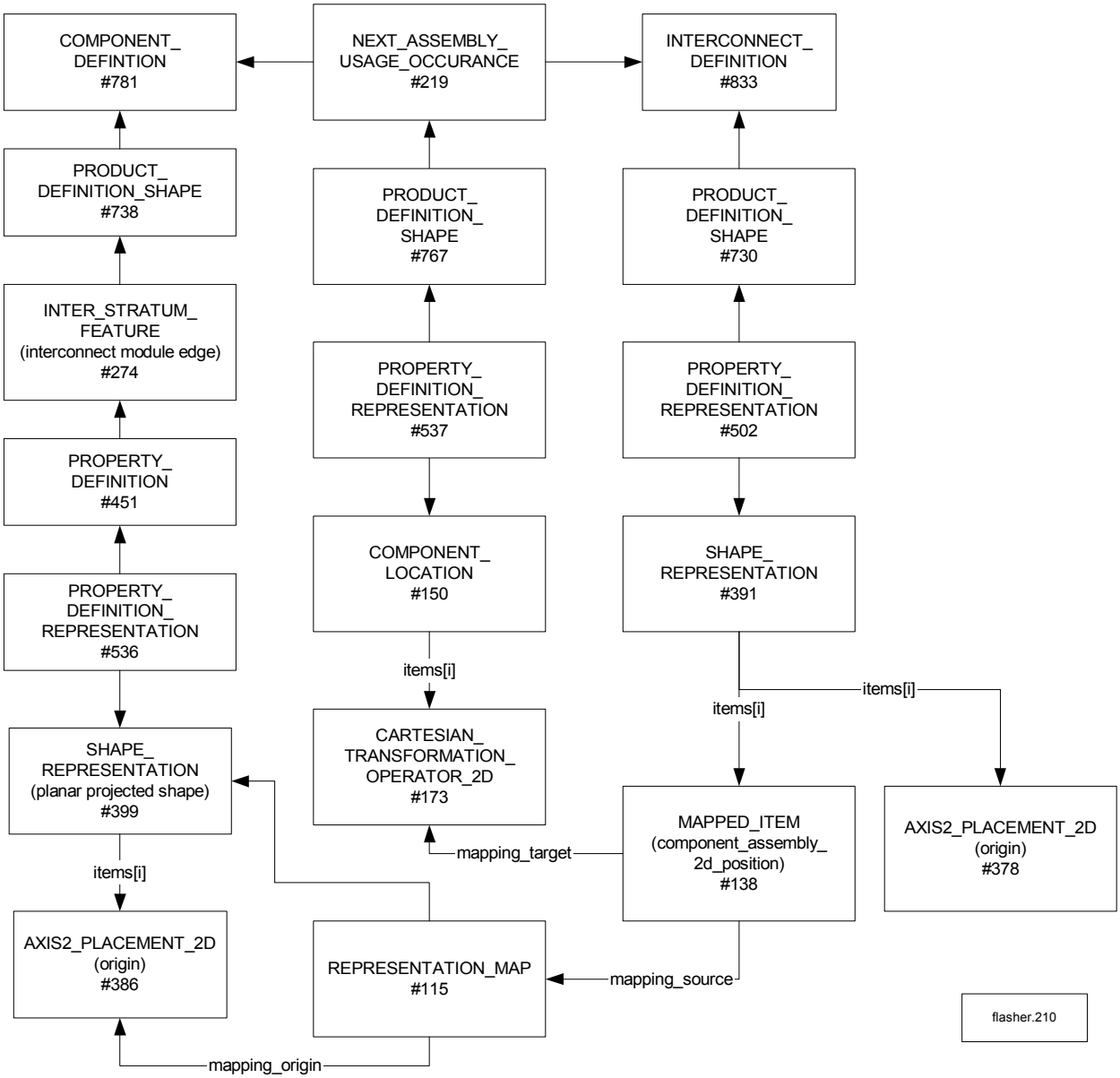


FIGURE 18 **Interconnect Module Edge Geometry**

- **Sample Part 21 Code**
  - **Physical Design view of PCB**

```
#83=INTERCONNECT_DEFINITION('','interconnect module',#80,#21);
#112=PROPERTY_DEFINITION('','',#83);
#111=PROPERTY_DEFINITION_REPRESENTATION(#112,#105);
#105=GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
'planar projected shape',(#106,#316,#322,#328,#334,#340,#346,#352,#358),
#39);
```

  - **Component Location**

```
#258=NEXT_ASSEMBLY_USAGE_OCCURRENCE('','assembly composition','',#83,#123,$);
#123=COMPONENT_DEFINITION('','interconnect module edge',#80,#15);
#320=PRODUCT_DEFINITION_SHAPE('','',#258);
#321=PROPERTY_DEFINITION_REPRESENTATION(#320,#294);
#294=COMPONENT_LOCATION('planar projected shape',(#295,#285,#316),#39);
#295=DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#285=CARTESIAN_TRANSFORMATION_OPERATOR_2D('','','',$,$,#276,1.);
#316=MAPPED_ITEM('component assembly 2d position',#317,#285);
#317=REPRESENTATION_MAP(#192,#189);
#192=AXIS2_PLACEMENT_2D('origin',#193,$);
#189=GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
'planar projected shape',(#192,#364),#39);
#267=SHAPE_DEFINITION_REPRESENTATION(#122,#189);
#122=PRODUCT_DEFINITION_SHAPE('','',#123);
```

  - **Units**

```
#39=(
GEOMETRIC_REPRESENTATION_CONTEXT(2)
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#48))
GLOBAL_UNIT_ASSIGNED_CONTEXT((#41,#45))
REPRESENTATION_CONTEXT('','')
);
#45=(
CONVERSION_BASED_UNIT('mil',#47)
LENGTH_UNIT()
NAMED_UNIT(#46)
);
```

The **component_definition** has a **product_definition_shape** that is an **inter_stratum_feature** with a description of “interconnect module edge”. If the shape is planar it represents a projection of the maximum extent of the Interconnect_module_edge onto the planar shape of the stratum.

2.5.3.3 The stratum entity

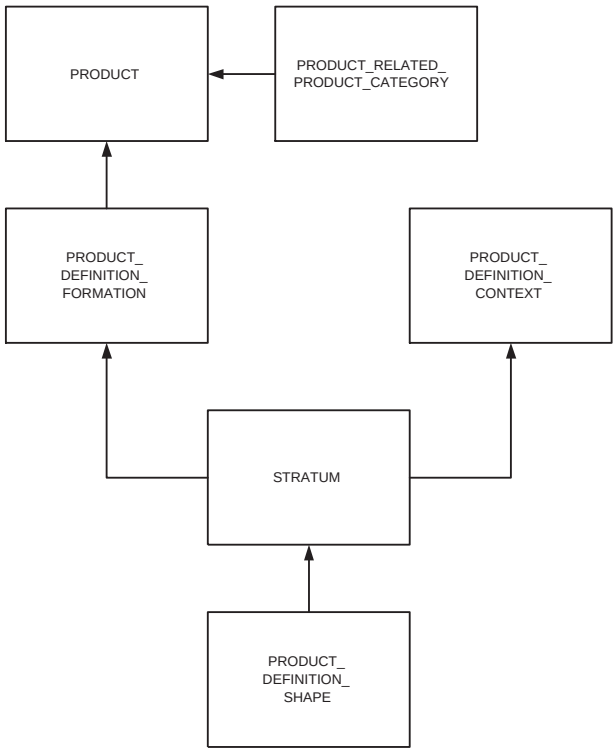


FIGURE 19 Stratum Definition

AP210 specifics the individual layers in the PCB AO using **stratum** entities which may be Design_layer_stratum AO or Documentation_layer_stratum AO or may be Stratum AO. **Stratum_feature** entities define the shapes in the stratum's material by referencing a **product_definition_shape** entity that references the stratum. This recommended practice is limited to describing planar stratum.

2.5.3.3.1 The Design Layer Stratum

A Design_layer_stratum AO is a type of stratum incorporated into an Interconnect_module for the purpose of realizing a physical network in one material such as signal, power and ground layers. A Design_layer_stratum AO is defined by a **stratum** entity with a formation attribute pointing to a **product_definition_formation** entity. The product_definition_formation's of_product attribute points to a **product** entity that is referenced by the products attribute of a **product_related_product_category** entity with a name attribute of "design layer".

2.5.3.3.2 The Documentation Layer Stratum

A Documentation_layer_stratum AO is a type of stratum used to describe patterns and other artwork information needed to manufacture a Stratum from materials which are not independently implementing a Physical_network in the PCB AO, such as silk screens and solder paste. A Documentation_layer_stratum AO is defined by a **stratum** entity with a formation attribute pointing to a **product_definition_formation** entity. The product_definition_formation's of_product attribute points to a **product** entity that is referenced by the products attribute of a **product_related_product_category** entity with a name attribute of "documentation layer".

2.5.3.3.3 The Stratum

A Stratum AO is used where there is no pattern information provided. A typical example is for the partially cured adhesive which reflows during fabrication to fill voids between Stratum_features in other Stratum.

2.5.3.4 The Stratum_feature entity

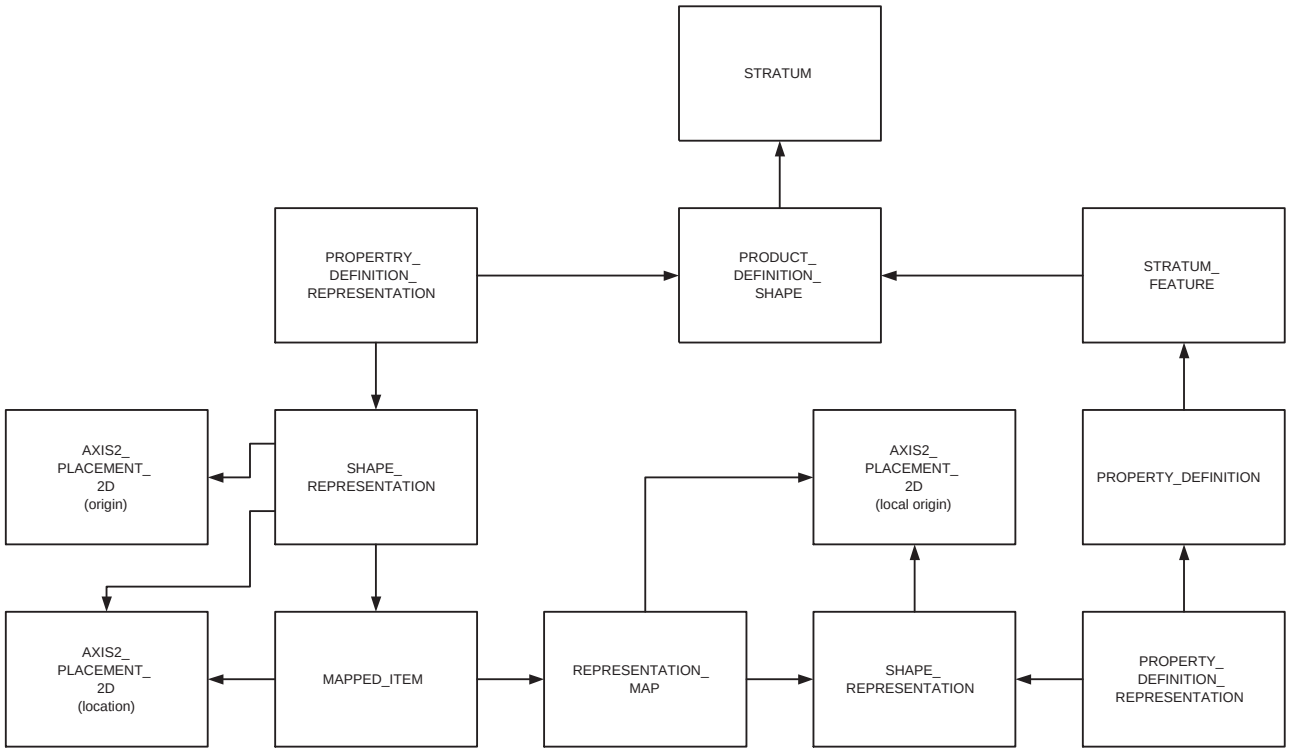


FIGURE 20 **Stratum Feature**

A **stratum_feature** entity is a subtype of shape_aspect that defines a shape realized in the material of the Stratum. The **stratum_feature** entity is associated with the stratum by its of_shape attribute with references a **product_definition_shape** entity whose definition attribute references the **stratum** entity. The **stratum** entity is referenced by the definition attribute of a **property_definition** entity. This property_definition is referenced by the definition attribute of a **property_definition_representation** entity. This property_definition_representation's used_representation attribute references a **shape_representation**[1] entity. The shape_representation defines the shape of the stratum_feature.

2.5.3.5 Conductive_interconnect_element

1. Even though the mapping table doesn't require it, the representation for the **stratum_feature** should be a **shape_representation**.

2.6 Integrating 2D and 3D CAD Layout Systems Using AP 210

This section describes how to formally map between 2D and 3D definitions within the context of AP 210 and how to convert between an AP 210 definition and a native CAD/CIM design/library definition. This mapping is typically used to support an integrated 2D/3D electrical-mechanical design and manufacturing library architecture. The model mapping technology is applicable to CAD/CIM environments although for brevity only the CAD acronym is used. The intended purpose is to support the definition of a component from the manufacturer of that component to the customer who receives that component installed in a higher level assembly from a different manufacturer. This purpose includes the system perspective that often the component manufacturer defines the component in one representation for use in a CAD system and delivers the physical component in a tape for assembly purposes. Simple coordinate transformations suffice to map one context onto another so that end to end simulation may be achieved.

2.6.1 AP 210 Reference CAD Layout Model

AP 210 provides a reference CAD model. The reference CAD model includes a 2D model, a 3D model, a feature model, a Datum Reference Frame model, and mapping models. AP 210 uses the STEP paradigm of explicit relationship assignment to implement the model. The features and Datum system definition are invariant between 2D and 3D representations. The features are related to the Datum system components explicitly.

The AP 210 reference CAD model supports physical package alterations by the enterprise (shape, lead trimming, etc.) (as compared to the as received package from the supplier) with the ARM Application objects Altered_package and Altered_packaged_part. Alterations that result in Form, or Fit changes are supported. For the purposes of this document, Form means the Geometric shape without tolerance data. Fit means the Geometric shape with tolerance data. Function means changes not included in either Form or Fit, typically identification, classification or behavioral changes. This section does not address behavioral modeling other than Fit.

2.6.1.1 Feature model

As described in previous sections of this document, AP 210 is features based. This allows persistent feature identification. The invariance of feature identification between different geometric representations and between data source and sink over the product life cycle is fundamental to successful usage of AP 210. The features in AP 210 are a common context for the geometric models and for the Datum Reference Frame models. The feature model is the link to network based definition and simulation systems.[1]

2.6.1.2 Datum Reference Frame model

FIGURE 21 illustrates a Datum Reference Frame. AP 210 provides a datum reference frame for the Package Application object based on combinations of reference planes and axes for a total of three Datum items. There should be at least one Datum_plane. Examples of Datum Reference Frame use are provided in later sections in this clause. The Datum reference frame model is also a link for

1. The standard has extensive feature definition capability and should be reviewed for details.

integrating electrical and mechanical analysis applications, but scenarios describing that usage is outside the scope of this section.

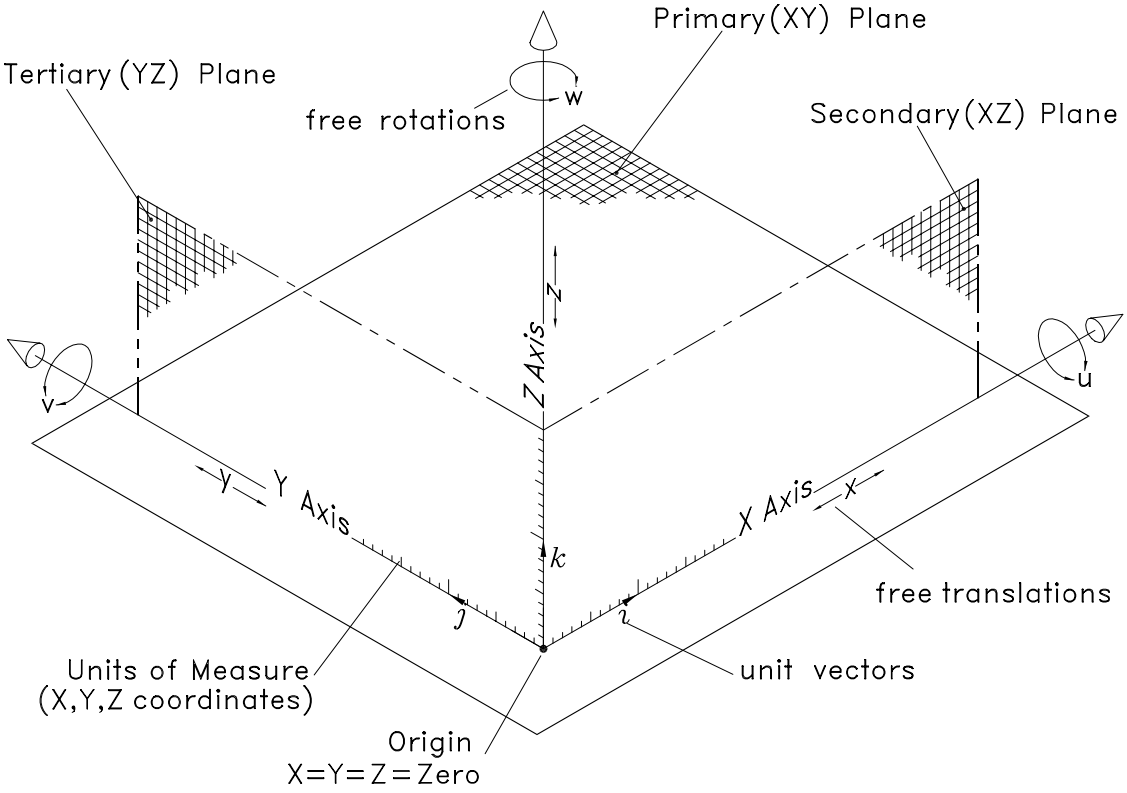


FIGURE 21 Datum Reference Frame

2.6.1.3 Connection_zone

The standard includes the ARM Application object Connection_zone which provides the shape of the active area on the terminal suitable for electrical or mechanical connection. The shape is provided in a geometric rather than a parametric context, so is not invariant between 2D and 3D representations, (e.g., a surface in the 3D representation on a rectangular cross-sectional terminal will be represented as a rectangle in 2D, but there is no guarantee that the real terminal has a connection zone at any place other than the intersection with the interconnect substrate). The standard provides parametric data to support design for manufacturing in that the Package Application object supports least_lead_length_below_seating_plane and maximum_lead_length_below_seating_plane properties for through hole technologies. The Package_terminal Application object supports minimum_terminal_extent and maximum_terminal_extent which are measured along an axis orthogonal to the centerline axis of the terminal at the seating plane. The extents provided in the Package_terminal Application object support correlation with the data in the ARM Application object Default_component_termination_passage_definition.

2.6.1.4 Library and Design Contexts

The AP 210 model architecture is based on the paradigm of separating the definitional and instantiation aspects of design to as great an extent as possible. The definitional aspects are

encompassed under the library context, while the instantiation aspects are encompassed under the design context. The integration mappings between 2D and 3D systems are maintained in the library context.

2.6.1.5 Concepts common to 2D and 3D models

The following concepts are in common between the 2D and 3D models:

Right handed coordinate system for the assembly definition,

Right handed coordinate system for the library part definition,

Right handed coordinate system for the interconnect definition.

The AP 210 reference CAD model has the following library properties defined in clause 4.2 of AP 210 for orientation purposes:

- Package.seating_plane,
- Volume_shape_projection.seating_plane,
- Primary_orientation_feature,
- Primary_reference_terminal,
- Polarity_indication_feature.
- Datum_point,
- Datum_plane,
- Datum_axis.

The AP 210 reference CAD library model is founded in a 3D coordinate system, in which a representation of the Datum reference system and representations of the appropriate features are placed. In many cases, the geometry in the shape representation in the 2D case is not necessarily a view as derived, e.g., from a camera model, but includes shapes that represent features necessary to satisfy assembly, analysis, layout, and inspection requirements. In particular an AP 210 2D library part shape should include geometry that represents certain features regardless of whether they are actually visible from the top of the component. An AP 210 2D library part shape should include items that represent certain Datum for orientation purposes. Explicit feature identification and explicit feature association with the individual geometric items in the shape_representation are provided. AP 210 requires Datum_axis population and Datum_plane population (ARM concepts) to ensure consistent mapping between the 2D representation and the 3D representation when coming in and out of CAD systems. It is essential to recognize that an enterprise using a 2D or 3D CAD system may use symbols to represent the Datum_axes and Datum_plane in cases where the intrinsic CAD functionality does not support these concepts. In these cases, the translators into and out of the AP 210 model must support the conversion of the symbols into the required data, which will include population of the datum concepts and the associated placement data.

The AP 210 reference CAD model includes the following properties in the ARM Application object Physical_unit_planar_shape and Physical_unit_3d_shape that are common to the 2D to 3D models:

- shape_purpose,
- shape_material_condition.

The relevant shape_purpose is design.

The relevant shape_material_condition are either: nominal_material_condition, maximum_material_condition, least_material_condition.

AP 210 supports multiple interconnect substrates in an assembly. In order to determine what substrate surface a component is mounted on, certain information must be provided. It is theoretically possible to derive which substrate is used to provide mechanical support for which component based on the connections from the component to the substrate and the geometric relationship between that substrate and that component. This approach may lead to model quality problems since it depends on evaluating geometric properties, and evaluation results may vary. AP 210 provides support for explicit relationships between the mating features to avoid model quality problems and to support evaluation of manufacturability. The AP 210 reference CAD model contains the ARM Application object Assembly_joint that specifies the component features joined by the manufacturing assembly process. This information is invariant among all types of 2D and 3D representations, and is a key determinant in establishing physical assembly relationships between features.

The AP 210 ARM Application objects Altered_package, Altered_packaged_part, and Altered_package_terminal parametric definitions are invariant among all types of 2D and 3D representations.

The AP 210 reference CAD model for the Application objects Component_2d_location Component_3d_location have an attribute mounting_surface to facilitate component placement in an assembly operation. The Component_feature referenced by the mounting_surface attribute is invariant between the 2d and 3d location for the same instance of Next_higher_assembly_relationship. This attribute is not as detailed as the ARM Application object Assembly_joint since it applies to the entire component.

Since the mounting_surface attribute references a Component_feature which is a generic instance concept, it is necessary to constrain in the standard the allowed subtypes of Part_feature that the component_feature may actually reference as a definition. The mounting surface identification information is conveyed by populating an instance of **component_shape_aspect** with the **description** of "interconnect module component surface feature". The mapping table provides a standard mapping to support this population.

The **shape_aspect_relationship** that specifies the **component_shape_aspect** by the **related** attribute shall have a **name** of "instantiated feature". The **shape_aspect** specified by the **relating** attribute of that **shape_aspect_relationship** shall be a **part_mounting_feature** with a **description** attribute value of "interconnect module primary surface" (corresponding to "top") or shall be a **part_mounting_feature** with a **description** attribute value of "interconnect module secondary surface".

The AP 210 reference CAD model attribute mounting_surface for the Application object Component_2d_edge_location (inherited from the supertype Component_2d_location) is required to be defined by an Interconnect_module_edge_segment_surface_feature. For orientation purposes, the Application object Component_2d_edge_location uses the instance of the Application object Assembly_joint referenced by Component_2d_edge_location.reference_terminal_assembly_joint to identify the top or bottom surface feature of the interconnect substrate that the Primary_reference_terminal of the component is assembled to. These relationships are invariant between the 2D and 3D CAD representations.

2.6.1.5.1 component to substrate distance constraint specification

The explicitly defined geometric interface between the packaged part and the substrate surface is the package seating plane. The role of the seating plane is to simulate the mounting surface in the assembly. The Package Application object contains sufficient parametric information to derive the maximum and minimum distance constraint between the substrate and the component body. The value derived from relative position measurements in the 3D explicit shape model shall be considered to be an approximation to the parametric data in the Application object. Enterprises may levy additional quality constraints beyond the length_uncertainty and angular_uncertainty included in the Cartesian_coordinate_system Application object.

2.6.1.6 AP 210 reference 3D model

The AP 210 reference 3D library context includes the following properties in the ARM Application object Physical_unit_planar_shape that are relevant to the 2D to 3D conversion topic:

- centroid_location,
- shape_purpose,

The relevant shape_purpose is design.

The location and orientation of a feature in the 3D library context is accomplished solely with the **axis2_placement_3d**, **mapped_item**, and **representation_map** entities. The location and orientation of a Connection_zone in the 3D library context is accomplished solely with the **axis2_placement_3d**, **mapped_item**, and **representation_map** entities. The graphic in the reference file ***package_terminal_3d.pdf*** illustrates the 3d definition of the feature subtype Package_terminal and of Connection_zone. The Datum_plane should be represented with a **plane**. The primary and secondary Datum axes should be represented with unit **vectors**. If the primary Datum_axis **vector** is perpendicular to the Datum_plane, it should originate at the Datum_plane and should be parallel to the Datum_plane normal. The secondary Datum_axis **vector** should originate at the Datum_plane and should be parallel to the normal of the Datum_plane.

The location and orientation of a component in the 3D assembly context is accomplished solely with the **axis2_placement_3d**, **mapped_item**, and **representation_map** entities. FIGURE 22 illustrates the 3D surface of the substrate.

The location and orientation of a component in the 3D interconnect context is accomplished solely with the **axis2_placement_3d**, **mapped_item**, and **representation_map** entities.

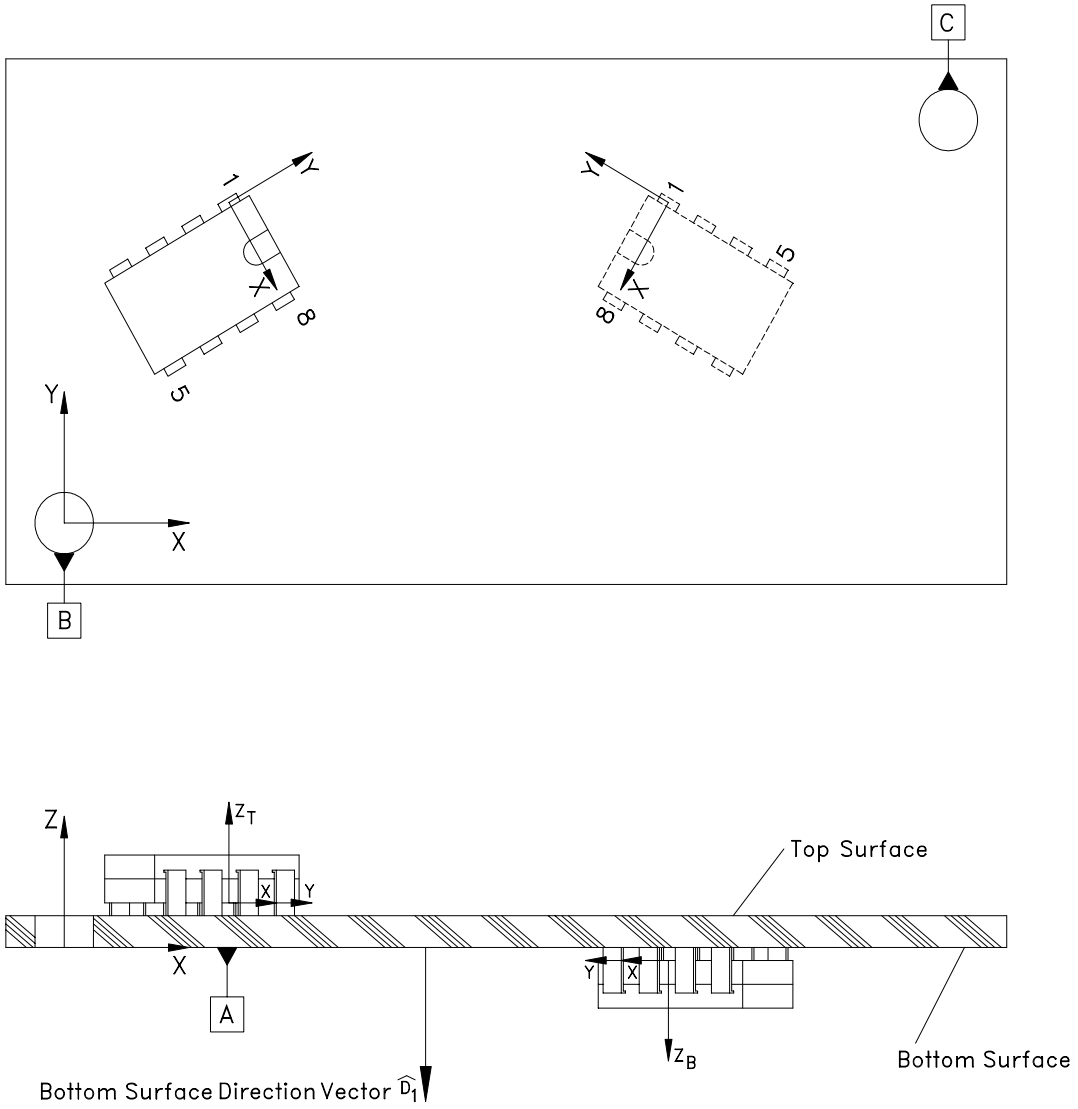


FIGURE 22 Three Dimensional Reference CAD Assembly Layout Model

2.6.1.7 AP 210 reference 2D model

The assembly coordinate system for a 2D context is always the "top" view[1].

1. The meaning of the word "top" in the AP 210 reference 2D model is that a "top" view is established by a plane that intersects the z axis of the coordinate system at a positive displacement from the zero of the z axis of the coordinate system, and the normal of that viewing plane is parallel to the normal of the x y (z = 0) plane of the coordinate system. This is the usual industrial usage of the "top view" term.

The interconnect coordinate system for a 2D context is always the "top" view.

The default library coordinate system for a 2D context is the "top" view.[1]

The location and orientation of a feature in the 2D library context is accomplished solely with the **axis2_placement_2d**, **mapped_item**, and **representation_map** entities. The location and orientation of a Connection_zone in the 2D library context is accomplished solely with the **axis2_placement_2d**, **mapped_item**, and **representation_map** entities. The Datum_plane is the primary member of the Datum Reference Frame and should be represented with the **axis2_placement_2d** that also represents the package origin. The secondary and tertiary Datum axes should be represented with either a **cartesian_point** or a unit **vector**. If the Datum_axis is perpendicular to the Datum_plane, it should be represented by a **cartesian_point**. If the Datum_axis is parallel to the Datum_plane, it should be represented by a **vector**. The Datum_axis related to the Primary_reference_terminal is a tertiary axis in the Datum reference frame and is required to be perpendicular to the seating plane.

In many cases, the geometry in the shape representation in the 2D case is not necessarily a view as derived, e.g., from a camera model, but includes shapes that represent features necessary to satisfy assembly, analysis, layout, and inspection requirements.

The library context includes the following properties in the ARM Application object Physical_unit_planar_shape that are relevant to the 2D to 3D conversion topic:

- centroid_location,
- shape_purpose,

The relevant shape_purposes are:

- design,
- design_profile,
- design_profile_above_seating_plane,
- design_profile_below_seating_plane
- seating_plane_based_package_shape,
- part_feature_viewing_plane_based_package_shape.

1. Part feature representations created for a special purpose (e.g., for representing an antenna interface of a transmitter module) may not be parallel to the seating plane.

All shape purposes other than part_feature_viewing_plane_based_package_shape are required by the standard to be parallel to the seating plane.

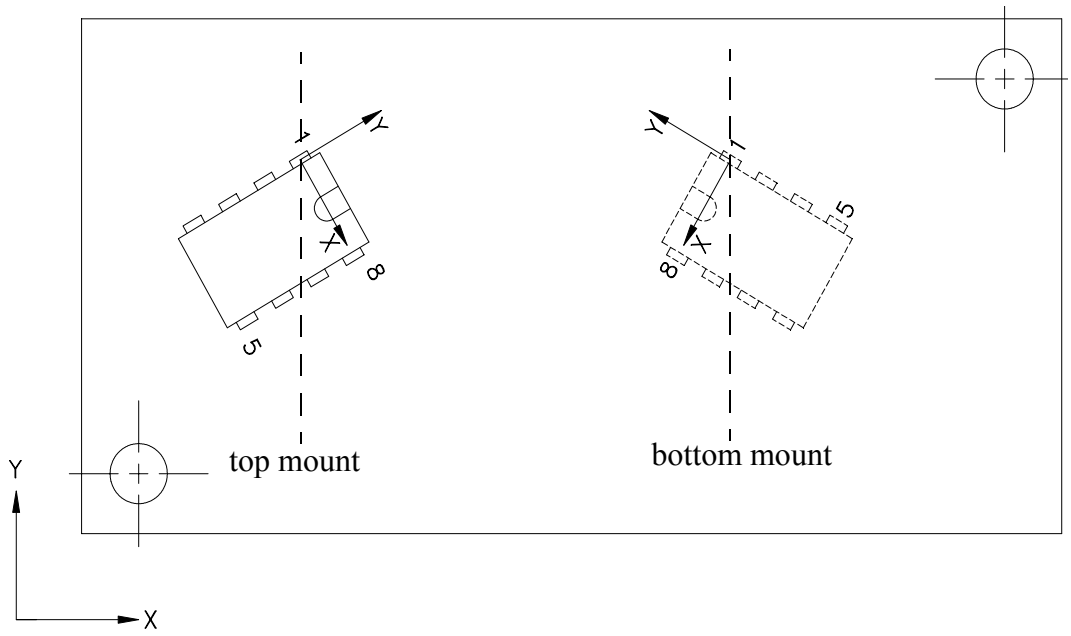


FIGURE 23 2D CAD Reference Assembly Layout model

The placement of a component when the component is mounted on the secondary surface of the substrate is illustrated in FIGURE 23 2D CAD Reference Assembly Layout model. The location and orientation of a component in the 2D assembly and interconnect models is accomplished solely with the **cartesian_transformation_operator_2d**, **mapped_item**, and **representation_map** entities. In this usage, **cartesian_transformation_operator_2d** is essentially considered to be a subtype of **axis2_placement_2d**, since the only additional allowed data is the mirroring information for the graphic. The scale attribute of **cartesian_transformation_operator_2d** is equal to 1.0. The determinant |T|, of the transformation matrix of the **cartesian_transformation_operator_2d** is equal to -1.0 when the component is mounted on the secondary surface of the substrate. The determinant is equal to 1.0 when the component is mounted on the primary surface of the substrate. Both axis1 and axis2 are required in AP 210 so that the determinant may be evaluated. The mirroring required to support this shall be about an axis that is parallel to the Y axis of the interconnect and that intersects the placement origin of the component. The samples in Figure 24 are offset for presentation purposes but Figure 24 is the 2D representation of the model represented in 3D in Figure 23. Figures 23 and 24 are simplified to focus on the library data. In a real design, the location transformation for the interconnect instance in the assembly would be included in order to derive the transformation that places that component in the context of that interconnect.[1]

2.6.1.7.1 Composite package shape

Enterprises may choose to create 2D representations for assembly and inspection or other purposes that are a composite of profiles or cross-sections of a 3D model. The 2D model is considered to be a

1. In the context of AP 210, both the component and the interconnect are in the geometric context of the assembly model.

sampling of cross-sections of a reference 3D model. Each individual cross-section of each **shape_aspect** being represented is placed into its own 2D definitional representation and then transformed into a composite 2D representation using part of the original sample data. This allows the 2D representation to be a stand-alone model. The ARM Application object attribute Physical_unit_planar_shape.purpose should be equal to "package composite 2d shape" when populating this data.

2.6.1.7.2 Part feature viewing plane shape

Specific part features or **shape_aspect**s may be represented in their own 2D package shape representation for library exchange, configuration control and management purposes. The need for this occurs when the physical feature or marking is not directly mated to the interconnect substrate. This capability should not be used if the feature in question does intersect or otherwise mate with the interconnect substrate. The most important features represented as applications of this include the following:

- Primary_orientation_feature,
- Polarity_indication_feature,
- Part_mounting_feature,
- Part_tooling_feature,
- Package_terminals that do not intersect the seating plane.

The value of Physical_unit_planar_shape.purpose for this scenario is "part feature viewing plane based package shape". This capability will be used when it is described how to map between 2D CAD and AP 210. The only orientation recommended for the Datum_axis representing one of these features is either to be perpendicular to the seating plane or to be parallel to the seating plane.

2.6.1.8 AP 210 mapping model

The AP 210 mapping model employs instances of shape_aspect_relationship, representation_relationship_with_transformation, and item_defined_transformation to establish the necessary mathematical and structural relationships between the 2D and 3D models.

Note: The mapping model requires the existence of the 2D and 3D models.

2.6.2 3D to 2D conversion in an AP 210 library context

The values for the attributes of ARM Application object Physical_unit_planar_shape shall be derived from the values of the source Physical_unit_3d_shape as modified by the sections below.

2.6.2.1 Datum Reference Frame Representation conversion

The Datum_plane is the seating plane and the seating plane algorithm is referenced. If the primary Datum_axis is perpendicular to the seating plane it’s representation should be converted using the seating plane algorithm. If the primary Datum_axis is parallel to the seating plane it’s representation should be converted using the Part feature viewing plane algorithm. The secondary Datum_axis is always perpendicular to the seating plane so it’s representation should be converted using the seating plane algorithm.

2.6.2.2 Seating Plane based algorithm

The following algorithm should be used to transform a 3D package representation to a 2D representation whose purpose is a "seating plane based package shape". The ARM Application

object Physical_unit_planar_shape attribute purpose equals "seating plane based package shape" for this case. The transformation from the **axis2_placement_3d** that represents the seating plane placement in 3d to the **axis2_placement_2d** that represents the package origin in 2d is derived by requiring the resultant **axis2_placement_2d** location to be (0.0,0.0) and direction to be (1.0,0.0). Any features shapes that intersect the seating plane or any Datum axes that intersect the seating plane shall be transformed into a 2D representation by considering the seating plane as a sampling function. The file ***package_terminal_s_p_2d_3d.pdf*** contains the graphic illustrating the data population for Package_terminal as established by the algorithm. Extending this data population to Package_body under seating plane algorithm and other subtypes of Package_terminal is straightforward.

2.6.2.3 Composite package shape algorithm

This algorithm applies when cross-sections of the package body (or package body and other features) are to be represented in one 2d shape. The 2d composite package shape is a separate 2d shape from that of the seating plane related shape and is used for other purposes. There are no constraints on the type of 2d geometry and against intersecting lines. It is recommended to use only closed curves and to use wireframe with topology to represent the 2d geometry to make it easier to accomplish extrusions in the inverse algorithm. The mapping repetitively accumulates samples of the 3d shape into an equivalent 2d shape, preserving orientation and location information. It is recommended that scaling be set to 1. Since this is a generic sampling algorithm, any 3D representation resulting from an inversion of this algorithm should be carefully reviewed to validate that an application of the algorithm meets enterprise requirements[1]. The instance data references identifiers found in the package body 2d - 3d shape_representation_relationship diagram in the file ***package_composite_2d_3d.pdf***

The algorithm:

```
 Create or identify a 3d package shape rep (#302)
 Create the 3d package body definitional shape rep (#600)
 Place the 3d package body shape rep in the package 3d shape rep using axis
placement, mapped item ...(#312, #310, #311)
   Create the first package body profile viewing plane axis placement at the
package level (#34201)
   Create the package body 2d shape rep (at the package level)(#110110)
   For each package body profile viewing plane axis placement at the package level
(#34200 or #34201):
     Sample #312 related shape in #302 using (#34200 or #34201)
     Create instance placement (#312222)
     Create profile shape rep (#1300 or #13100)
     Create mapped item and rep map to connect #312222 and (#1300 or #13100)
     Create item defined transformation and rep rel with transf {(#13202,#13203),
(#13303, #13303)} to connect (#34201, #34200) to #110110 to establish the
placement constraints on the second and later profiles.
 Endfor
End Algorithm
```

[1] Accurate profile generation can be supported by the data defined by Conformance Class 11 of the standard, Geometric Dimensioning and Tolerancing.

2.6.2.4 Part feature viewing plane shape

The value of Physical_unit_planar_shape.purpose for this scenario is "part feature viewing plane based package shape". The same algorithm is used as above to extract the 2D shape, except that there is only one sample plane. The primary purpose of this transformation is to create a separately identifiable 2D representation for the specific Part_feature. The viewing plane is used as a sampling device in order to create a definitional 2D shape representation attached to the feature and also to a 2D shape representation attached to the package. When there is only one viewing plane, it is recommended that the viewing plane normal be the **vector** representing the part feature axis. The 2D representation should then also be a **vector**. The file ***part_feature2d_3d.pdf*** contains the graphic illustrating the data population for Package_terminal as established by the algorithm.

2.6.3 3D to 2D conversion in an AP 210 assembly context

The **cartesian_transformation_operater_2d** supporting component_location is derived from the **axis2_placement_3d** of the **next_assembly_usage_occurrence** of the **component_definition** and the **axis2_placement_3d** of the **next_assembly_usage_occurrence** of the Interconnect_module_component in conjunction with the mounting_surface attribute value. The scale value of the **cartesian_transformation_operater_2d** shall equal 1.0.

2.6.3.1 top or bottom surface

The determinant |T|, of the transformation matrix shall be constrained to be equal to -1.0 if the mounting surface is the secondary surface. The determinant |T|, of the transformation matrix shall be constrained to be equal to 1.0 if the mounting surface is the primary surface.

2.6.3.2 edge surface

The mounting_surface shall be provided. The determinant |T|, of the transformation matrix shall constrained to be equal to -1 if the Primary_reference_terminal of the Packaged_connector is assembled to the secondary surface of the interconnect substrate. The determinant |T|, of the transformation matrix shall constrained to be equal to 1 if the Primary_reference_terminal of the Packaged_connector is assembled to the primary surface of the interconnect substrate. The assembled surface for the Primary_reference_terminal is determined by a query on the attribute Component_2d_edge_location.reference_terminal_assembly_joint.assembly_features[i].

2.6.4 3D to 2D conversion in an AP 210 interconnect context

The **cartesian_transformation_operater_2d** supporting component_location is derived from the **axis2_placement_3d** of the **next_assembly_usage_occurrence** of the **component_definition** in the Interconnect_module design definition 3D representation. The determinant |T| of the transformation matrix value shall be 1.0 in all cases. Since the topography of the design does not change between 3D and 2D, the transformations are trivial. Suggestions for algorithms to convert non-planar surfaces into a 2D system is outside the scope of this document.

2.6.4.1 inter_stratum_feature

There are no special considerations for this case. The context for the inter stratum feature is that of the entire interconnect substrate definition. The inter stratum extent is defined by an geometric context

invariant population of **shape_aspect_relationship**. All Inter_stratum_features are represented by closed curves.

2.6.4.2 stratum_feature_template_component

There are no special considerations for this case. The context for the stratum_feature_template_component is that of the entire interconnect substrate definition.

2.6.4.3 stratum_feature

There are no special considerations for this case. The shape of the stratum feature is defined by a collection of primitive shapes and operations on those shapes. The geometric context for the stratum feature is that of the entire interconnect substrate definition.

2.6.4.4 stratum

There are no special considerations for this case. The shape data for a stratum in 3D is a solid. In 2D the shape representation is a geometrically bounded wireframe or basic curves. In creating the 2D model, the conversion application should ensure that all **curve**s representing **stratum** boundaries are closed.

2.6.4.5 Interconnect substrate thickness

The conversion application should ensure that the solid model is consistent with the board thickness parameter found in the ARM AO Interconnect_module_usage_view.

2.6.5 2D to 3D conversion in an AP 210 library context

The values for the attributes of ARM Application object Physical_unit_3d_shape shall be derived from the values of the source Physical_unit_planar_shape as modified by the sections below.

2.6.5.1 Datum Reference Frame Representation conversion

The Datum_plane is the seating plane and the seating plane algorithm is referenced. If the primary Datum_axis is perpendicular to the seating plane it's representation should be converted using the seating plane algorithm. If the primary Datum_axis is parallel to the seating plane it's representation should be converted using the Part feature viewing plane algorithm. The secondary Datum_axis is always perpendicular to the seating plane so it's representation should be converted using the seating plane algorithm.

2.6.5.2 Seating plane based case

The following algorithm should be used to transform a 2D representation whose purpose is a "seating plane based package shape" to a 3D representation when no 3D representation exists. The ARM Application object Physical_unit_planar_shape attribute purpose equals "seating plane based package shape" for this case. This representation should be extractable from all 2D CAD systems used for layout because it is closely tied to the land pattern definition. The **axis2_placement_3d** that represents the package origin in 3d is derived from the **axis2_placement_2d** that represents the package origin in 2d by adding a z axis and copying attributes as defined in the equations below.

xis2_placement_3d.location.coordinates[1] = axis2_placement_2d.location.coordinates[1] **(1)**

axis2_placement_3d.location.coordinates[2] = axis2_placement_2d.location.coordinates[2] (2)

axis2_placement_3d.ref_direction.direction_ratios[1] = axis2_placement_2d.ref_direction.direction_ratios[1] (3)

axis2_placement_3d.ref_direction.direction_ratios[2] = axis2_placement_2d.ref_direction.direction_ratios[2] (4)

The **axis2_placement_3d** that represents the seating plane should be congruent with the **axis2_placement_3d** included to represent the package origin. Swept area extrusions should be used for the terminal features, where the extruded area is the feature shape in 2d, and the extrusion length L is

□ = Package.maximum_body_height_above_seating_plane + Package.maximum_lead_length_below_seating_plane (5)

where the package is a through hole package. For surface mount package terminals, use the package body thickness parameter T. The location of features and Datum in 3D should be accomplished similarly to the location of the origin. A swept area extrusion should be used for the package body, where the vertical thickness T is

p = Package.maximum_body_height_above_seating_plane – Package.minimum_body_clearance_above_seating_plane (6)

where the package body is entirely above the seating plane. Use the following value for T

p = Package.maximum_body_height_below_seating_plane – Package.minimum_body_clearance_below_seating_plane (7)

where the package body is entirely below the seating plane. Use the following value for T

p = Package.maximum_body_height_above_seating_plane + Package.maximum_body_height_below_seating_plane (8)

where the package body intersects the seating plane. All extrusions should be perpendicular to the seating plane.

Note: The supported usage of a seating plane is to allow component installation processes in manufacturing to understand which side of the substrate to approach from when installing a component. A component with a body that is below the seating plane in the "as built" configuration (i.e., in a cutout) is still considered to be installable from above the seating plane. Consequently, the Package_terminals are always installed from above the seating plane moving toward the seating plane.

After the geometry has been created, ensure that the product structure side of the model (subtypes of shape_aspect, shape_aspect_relationship, property_definition, property_definition_representation) are populated and the 3D geometric model is related to the product structure side of the 2D model per the standard requirements. The required instances of **representation_relationship_with_transformation** and **item_defined_transformation** should be populated. The referenced graphic illustrates the data populated to support a combined 2d / 3d library where the 2d representation is based on the component seating plane and where there is only terminal information in the 2d representation (i.e., no package body information in that 2d representation). ***package_terminal_s_p_2d_3d.pdf***.

2.6.5.3 Composite package shape

The 2D model is considered to be a sampling of cross-sections of a reference 3D model. Each individual cross-section of each **shape_aspect** being represented is placed into its own 2D definitional representation and then transformed into a composite 2D representation using part of the original sample data. An algorithm similar to that in the previous paragraph may be executed, but

consideration should be provided for sampling errors in constructing the 3D model. The method does not provide an explicit relationship between subsequent samples and relies on querying the vertical height data during conversion from 2D to 3D. The ARM Application object attribute Physical_unit_planar_shape.purpose should be equal to "package composite 2d shape" when generating this data. The referenced graphic illustrates some of the data populated to support this scenario. **package_composite_2d_3d.pdf**.[1]

2.6.5.4 Part feature viewing plane shape

Specific part features or **shape_aspect**s may be represented in their own 2D package shape representation for library exchange, configuration control and management purposes. The need for this occurs principally when the physical feature or marking is not directly mated to the interconnect substrate. In fact, this capability should not be used if the feature in question does intersect or otherwise mate with the interconnect substrate. The most important features represented as applications of this include the following:

- Primary_orientation_feature,
- Polarity_indication_feature,
- Part_mounting_feature,
- Part_tooling_feature,
- Package_terminals that do not intersect the seating plane.

The value of Physical_unit_planar_shape.purpose for this scenario is "part feature viewing plane based package shape". The same algorithm is used as above to create the 3D shape. As in the case of the composite shape, the primary purpose of this transformation is not to create an extrusion but to locate precisely the shape of the feature in the 3D model or to provide orientation information for integrating electrical, mechanical, and manufacturing engineering data. This capability will be used in later developments when it is described how to map between 2D CAD and AP 210. Another example of use would be for fiber-optic transmitters where the waveguide fiber optic interface is not parallel to the seating plane. Of course, there are countless cases of right-angle coaxial connectors. In these cases, the Part_feature would be a Packaged_part_interface_terminal and possibly a Guided_wave_terminal. Since there is no guarantee that features are at any specific angle with respect to the seating plane, when that information is required, **representation_relationship_with_transformation** and **item_defined_transformation** provide the information. The only orientation recommended for the Datum_axis representing these features is either to be perpendicular to the seating plane or to be parallel to the seating plane. In no case should the two Datum axes be congruent. The majority of cases do occur where the mating condition is at right angle to the seating plane because the module the PCA is a part of is intended to slide into the next higher assembly. The referenced graphic illustrates the data populated to support this scenario. ***part_feature2d_3d.pdf***

2.6.6 2D to 3D conversion in an AP 210 assembly design context

The **axis2_placement_3d** of the component location is derived from the **cartesian_transformation_operater_2d**, mounting_surface, board thickness, and board placement and orientation. In cases where components are mounted in cutouts, local thickness information derived from Inter_stratum_features (the cutouts) associated with the component may be necessary to

1. Accurate 3D reconstruction can be achieved by using profiling capability in Conformance Class 11: Geometric Dimensioning and Tolerancing.

generate an accurate 3D model. The file ***transformation_matrix2d3d.pdf*** is an explicit derivation of the values of **axis2_placement_3d** with an illustrated test case.

2.6.7 2D to 3D conversion in an AP 210 interconnect design context

The **axis2_placement_3d** of the component location is derived from the **cartesian_transformation_operater_2d**, **stratum** thickness, and **stratum** relative position with respect to the origin in the Z axis. The 3D solid model of the interconnect substrate is an extrusion based on the ARM AO Interconnect_module_usage_view and the population of Adjacent_stratum_surface_definition desired to be included in the 3D model. Normal industrial usage is to consider the origin of the interconnect substrate to be the first Design_layer_stratum. The first Design_layer_stratum is that Stratum that contains the Primary_stratum_indicator_symbol. There may be only one Primary_stratum_indicator_symbol in a design. Since there may be Stratum on either side of that Stratum, and since the thickness of Stratum is not considered in 2D CAD, assigning the Z axis zero location to the first Design_layer_stratum may lead to errors. It is recommended that the enterprise develop a consistent algorithm for converting the first Design_layer_stratum information into the origin of the 3D solid model. There is sufficient information included in AP 210 to avoid ambiguity once the data is in AP 210 context.

2.6.8 Conversion Between 2D CAD system and AP 210

This section includes a description of a common 2D CAD system capabilities and recommendations on how to incorporate certain concepts from AP 210 into that CAD system.

2.6.8.1 Deviation from strict GD&T naming convention

GD&T modeling technology is used in this application in the standard for part orientation information, not for manufacturing information. In a 2D application, it is understood and implicit that the package is to be mounted onto a planar surface, the interconnect substrate. Therefore, information about this critical surface is not explicitly stored or represented in 2D library data systems. In fact, little information about the package is typically stored in these libraries. Specifically information about the bottom surface that is the datum feature most often used to derive a datum plane from which the datum reference frame is constructed is not stored because the shape cannot really be represented (the outline of the package is not the shape of the bottom surface). A critical item of information that is storeable is the index mark found on many packages. The Electrical Engineering domain usage of the index mark is to consider that mark as a "primary orientation feature" and AP 210 has used that term in accordance with that usage to define the ARM Application object Primary_orientation_feature. Obviously, the index mark is not the bottom surface of the package either.

The Primary_orientation_feature may be assigned to a Datum_axis. It follows that if the primary datum must be a plane, the primary datum is not associated with the "primary orientation feature", but with a different Part_feature.

Note: In a similar fashion, information about a "secondary orientation feature" is storeable in 2D library data systems.

Therefore, applications will have to query the GD&T datum assignment to datum features in order to construct a datum reference frame instead of assuming the Primary_orientation_feature is always assigned to the primary datum plane.

It is recommended to use the tertiary_orientation_feature attribute of Package to identify the Part_feature that is the target for the primary datum plane assignment. This allows the Application objects Primary_orientation_feature and Secondary_orientation_feature to be explicitly represented in the 2D CAD library system.

In some cases, the component has partial symmetry and orientation is incompletely specified by the component manufacturer. In these cases, the library data should be considered as a defining data source due to the relationships established to the electrical functional model in the library (e.g., pin mapping).

Note: A consequence of this is that a drawing or illustration containing a Primary_orientation_feature identification may show datum "A" assigned to the bottom surface feature of a Package but the data set associated with that Package will state that the bottom surface Part_feature is in the role of tertiary_datum_feature for a Package. Consequently, in order to synchronize the drawing and the datum reference frame, a primary datum in a datum reference frame may be assigned to a feature which is in the role of secondary_orientation_feature or tertiary_orientation_feature.

Note: The Primary_orientation_feature is NOT necessarily associated with pin 1.

2.6.8.2 Sample 2D CAD system characteristics

This case study is based on common industry practice but is not necessarily representative of all systems. The 2d CAD system considered in this case supports mirroring as specified by the ISO 10303-42 entity **cartesian_transformation_operator_2d**. The CAD system considered in this case displays all data to the operator as viewed from the "top" of the substrate (including components that are mounted on the bottom of the substrate). The CAD system considered herein does know the side of the substrate the component is mounted on. The CAD system considered herein has one right handed coordinate system for combined interconnect / assembly design. The local component coordinate system is a right handed coordinate system. The library coordinate system is a right handed coordinate system. If terminal identifiers follow a counter-clockwise pattern of increasing value when the component is mounted on the top of the substrate, then the terminal identifiers following a clockwise pattern of increasing value when the component is mounted on the bottom of the substrate.

2.6.8.3 Seating_plane

The Seating_plane should be represented in a 2D CAD system by the Package origin symbol discussed below. The Seating_plane should always be at least partially derived from orientation features. The Seating_plane should be parallel with either one of the {X-Y, Y-Z, Z-X} planes of the datum reference frame. A Seating_plane may be congruent with a Datum_plane (e.g., in the case of an LCC package).

2.6.8.4 Parametric Data Required[1]

The following parametric data is required:

- Package.maximum_body_height_above_seating_plane.[2]

The following additional parametric data are required for certain packages:

1. Reference 2D to 3D conversion section for details.

2. The standard requires that this value represent the maximum height of the package (including terminals) above the seating plane.

- Package.maximum_body_height_below_seating_plane (for cutout mounted packages),
- Package.maximum_lead_length_below_seating_plane (for through hole packages).

The following additional parametric data are required for accurate bidirectional 2D - 3D conversion:

- Package.minimum_body_clearance_above_seating_plane,
- Package.minimum_body_clearance_below_seating_plane (for cutout mounted packages).

Other parametric data as described in the Package application object is unnecessary for the simple extrusion based conversion from 2D to 3D but is appropriate for more accuracy and for other tools (i.e. mechanical simulation).

2.6.8.5 Graphical Data Recommendations

This clause identifies the specific recommendations for graphical symbology and feature shape representations.

2.6.8.5.1 Primary Datum_plane

In this recommended practice, there is no separate symbol for the primary Datum_plane as such. In the 2D CAD system, the primary datum plane may be implicitly defined in the case where it is parallel to the seating plane (represented by the same Package origin symbol as the seating plane). In the case where it is not parallel to the seating plane, the location will have to be derived from the other datum feature information. If it is parallel to the seating plane, then the parametric information may be used to create the geometric representation (e.g., the bottom surface representation).

2.6.8.5.2 Primary orientation feature Datum_axis symbol

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the feature Datum_axis. There are two cases that can be supported with this method. The first case is where the feature Datum_axis intersects the seating plane. In this case, the symbol graphic should be a small circle centered on the axis. The second case is where the axis of the feature is parallel to the seating plane.

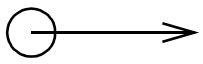


FIGURE 24 2D CAD Vector Symbol

In this case, the vector symbol graphic is composed of a circle, line and arrowhead should be used and is illustrated in FIGURE 24. The circle is centered on the terminus of the feature and the vector axis represents the Datum_axis of the feature. The intersection of the axis line and the polyline making up the arrowhead should be used for the second point defining the vector.

2.6.8.5.3 Primary orientation feature Datum_plane symbol

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the feature Datum_plane. This symbol represents the normal of the Datum_plane associated with the Primary_orientation_feature. The Datum_plane intersects the seating plane at 90 degrees. In this case, the symbol graphic should be as shown in the figure below. The plane shall be at the centreline of the thick line and the origin of the vector shall be at the intersection of the centrelines of the two displayed lines. The intersection of the axis line and the polyline making up the arrowhead should be used for the second point defining the vector..

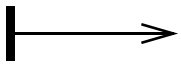


FIGURE 25 2D CAD Plane Based Vector Symbol

2.6.8.5.4 Primary orientation feature shape

It is recommended to include a layer in the 2D CAD system that captures the profile through the Primary_orientation_feature for documentation and illustration purposes. This will assist librarians in auditing decisions as to whether to use an axis or plane symbol and allow validation of symbol orientation. This shape is intended to be a partial cross-sectional shape of the feature. This shape shall be consistent with the Datum assigned to the feature. In many cases, this shape is an idealization of actual part marking or shape eccentricities intended to allow operators or machinery to orient the component in the installation process.

2.6.8.5.5 Secondary orientation feature Datum_axis symbol

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the feature Datum_axis. There are two cases that can be supported with this method. The cases are identical with the Primary_orientation_feature and the same treatment and graphic symbology should be used.

2.6.8.5.6 Secondary orientation feature Datum_plane symbol

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the feature Datum_plane. This symbol represents the normal of the Datum_plane associated with the Secondary_orientation_feature. The Datum_plane intersects the seating plane at 90 degrees. In this case, the symbol graphic should be identical to that for the Primary_orientation_feature plane case.

2.6.8.5.7 Secondary orientation feature shape

It is recommended to include a layer in the 2D CAD system that captures the profile through the Secondary_orientation_feature for documentation and illustration purposes. This will assist librarians in auditing decisions as to whether to use an axis or plane symbol and allow validation of symbol orientation. This shape is intended to be a partial cross-sectional shape of the feature. This shape shall be consistent with the Datum assigned to the feature. In many cases, this shape is an idealization of actual part marking or shape eccentricities intended to allow operators or machinery to orient the component in the installation process.

2.6.8.5.8 Primary reference terminal axis symbol

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the feature axis. It is recommended to use a small circle with the center of the circle at the axis of the feature. This allows the highest accuracy mapping between the native CAD application and AP 210. A circle is recommended because there will be no confusion about the orientation of the symbol itself adding orientation information. The axis of this feature at the interface to the seating plane is required to be parallel to the seating plane normal. The primary reference terminal is not required to be an orientation feature since it is usually indistinguishable from other terminals. If the primary reference terminal is also an orientation feature, then a complex instance of Primary_orientation_feature and Primary_reference_terminal shall be populated to indicate that situation.

2.6.8.5.9 Polarity indication feature symbol

The case considered is that there is a layer set aside in the 2D CAD system for a symbol that represents the feature. It is recommended to use a small circle with the center of the circle at the centroid of the shape representing the feature projected onto the seating plane. This allows the highest accuracy mapping between the native CAD application and AP 210. A circle is recommended because there will be no confusion about the orientation of the symbol itself adding orientation information. A projection is recommended because of the high frequency of axial parts with band feature shapes.

2.6.8.5.10 Polarity indication feature shape

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the shape of the polarity indication feature. This shape is recommended to be a projection shape of the feature. In many cases, this shape is an idealization of actual part marking or shape eccentricities intended to allow operators or machinery to orient the component in the installation process. The centroid of the bounding box for the feature shape should be congruent with the center of the circle that is the symbol for the feature.

2.6.8.5.11 Package origin symbol

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the origin axis. It is recommended to use a small circle with the center of the circle at the origin of the package. This allows the highest accuracy mapping between the native CAD application and AP 210. A circle is recommended because there will be no confusion about the orientation of the symbol itself adding orientation information.

2.6.8.5.12 Package X direction symbol

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the X direction. It is recommended to use a circle with the center of the circle on the positive X axis, and at a distance equal to twice the diameter of the circle representing the package origin from the origin. This allows the highest accuracy mapping between the native CAD application and AP 210. A circle is recommended because there will be no confusion about the orientation of the symbol itself adding orientation information. The circle should have a 50% smaller diameter than the circle representing the package origin for convenience when plotting test samples.

2.6.8.6 Orientation Case Examples

Some common cases are described, with graphic illustrations showing how the recommendations were applied. Composite figures are provided to illustrate in aggregate the usage of the recommendations. See Figures xx through yy.

2.6.8.6.1 Horizontal axial through hole non-polarized packages

The orientation features for this case are three cylinders, the body and the two terminal features. The body (a cylinder) is the Primary_orientation_feature. One of the terminals (also a cylinder) is the Secondary_orientation_feature. The primary Datum_plane is derived from the intersection of the axes for these two features. The Seating_plane X axis should be parallel to the Datum_axis associated with the Primary_orientation_feature. The Seating_plane is perpendicular to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature should intersect the Seating_plane at right angles. In order to fully orient the part, a tertiary orientation feature is required to intersect the Seating_plane also at a right angle, which role is satisfied by the Primary_reference_terminal. The selection of a Primary_reference_terminal is arbitrary since the

component manufacturer rarely specifies this information. In this case, the library data is the defining artifact for this information.

2.6.8.6.2 Vertical axial through hole non-polarized packages

The orientation features for this case are two cylindrical features and a surface feature. The primary Datum_plane is associated with the bottom surface of the body cylinder, which is the Primary_orientation_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles. In order to fully orient the part, a tertiary orientation feature is required, which role may be satisfied by the Primary_reference_terminal.

2.6.8.6.3 Horizontal axial through hole polarized packages

The orientation features for this case are two cylindrical features, the body and one terminal feature. The body (a cylinder) is the Primary_orientation_feature. One of the terminals (also a cylinder) is the Secondary_orientation_feature. The primary Datum_plane is derived from the intersection of the axes for these two features. The Seating_plane X axis should be parallel to the Datum_axis associated with the Primary_orientation_feature. The Seating_plane is perpendicular to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature should intersect the Seating_plane at right angles.

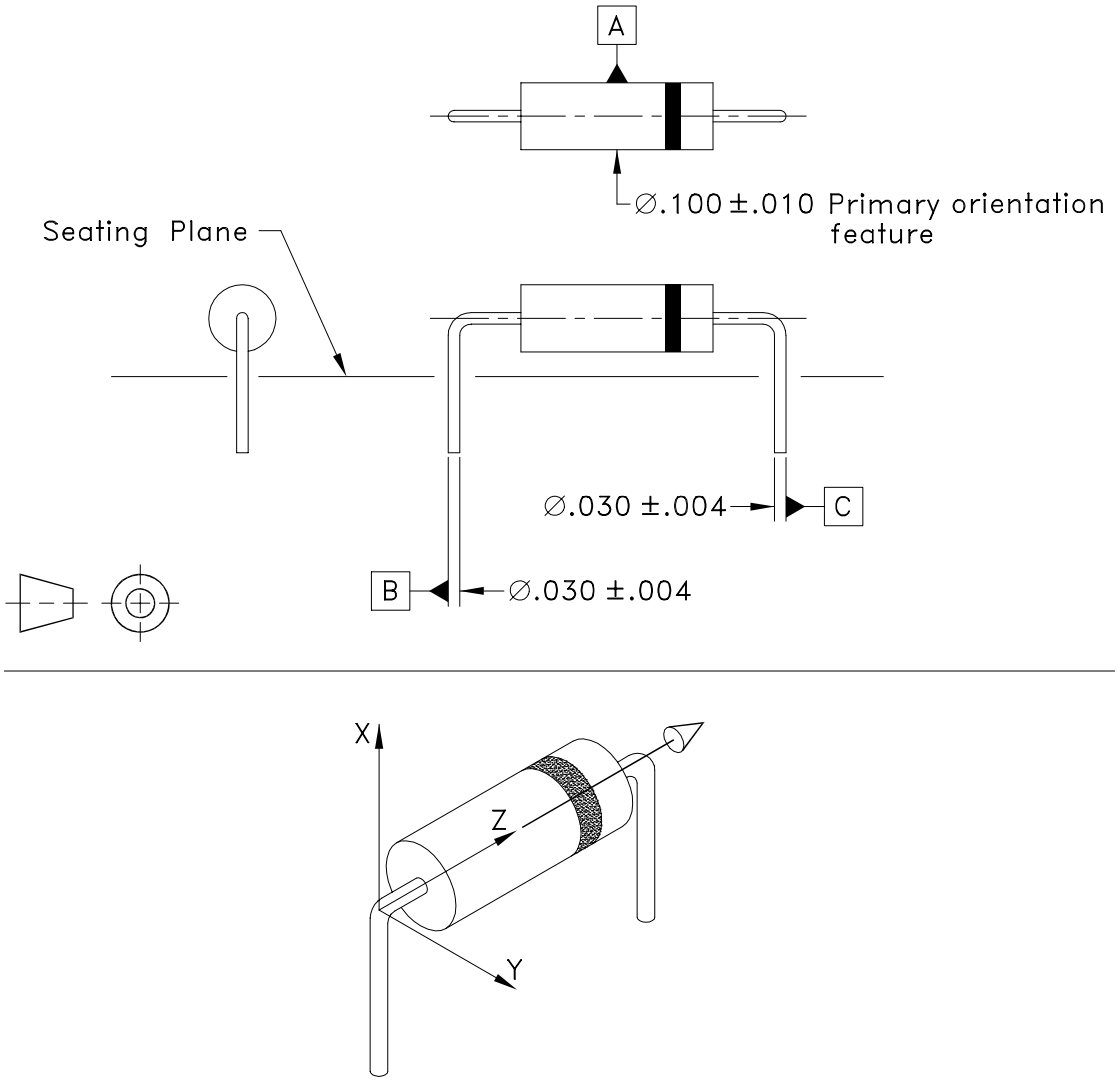
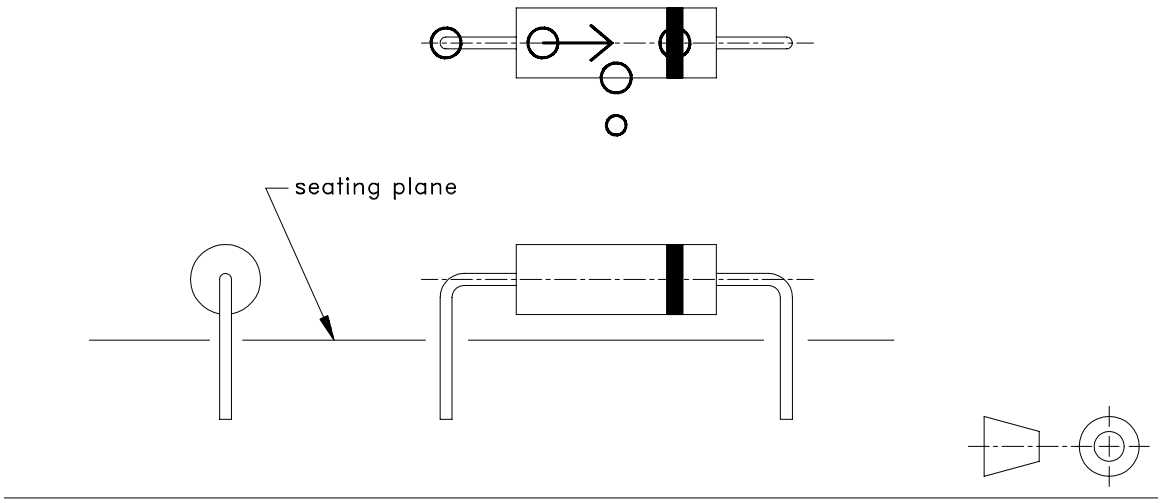


FIGURE 26 horizontal axial through hole polarized

FIGURE 29 Cyndrical through hole with orientation Symbology is an alternate representation of the part which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology.



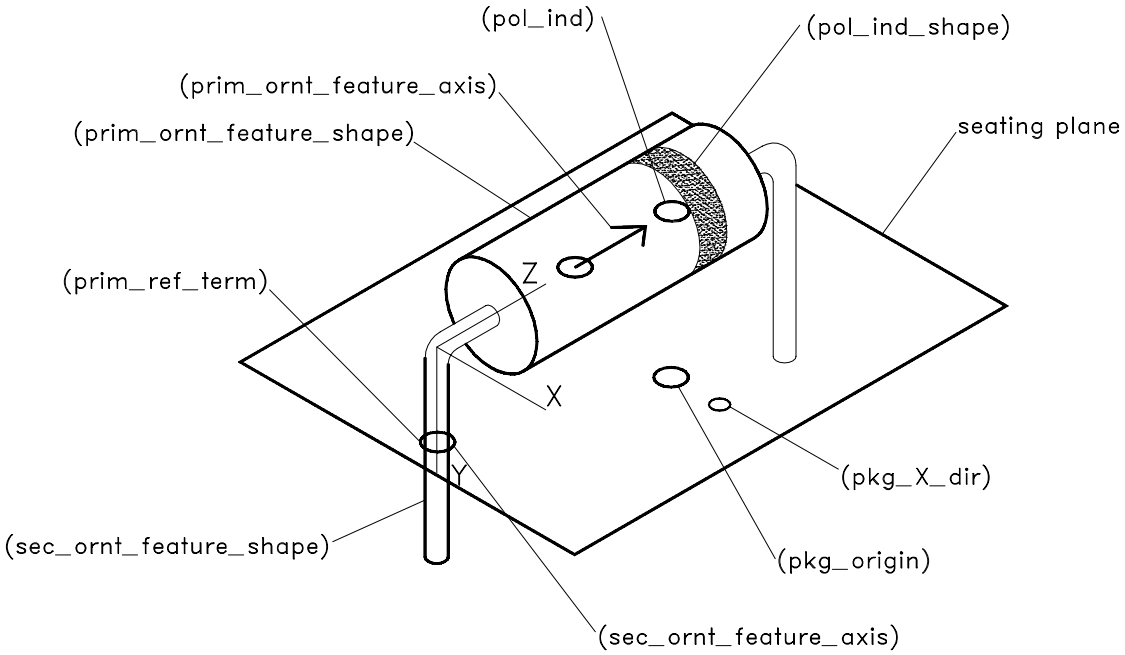


FIGURE 27 horizontal axial through hole polarized with orientation Symbology[1]

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

2.6.8.6.4 Vertical axial through hole polarized packages

The orientation features for this case are two cylindrical features and a surface feature. The primary Datum_plane is associated with the bottom surface of the body cylinder, which is the Primary_orientation_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

2.6.8.6.5 Rectangular through hole packages

The orientation features for this case are: an opposing side surface feature set, a bottom surface feature, and a dimple in the top surface. These features result in a plane, plane, axis datum reference system. The Primary_orientation_feature is the dimple in the top surface. The Secondary_orientation_feature is the opposing sides surface set. The tertiary_orientation_feature is the bottom surface. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature in the role of tertiary_orientation_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

2.6.8.6.6 Cylindrical through hole packages

The orientation features for this case is one cylindrical feature, one opposing plane feature, and a surface feature. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

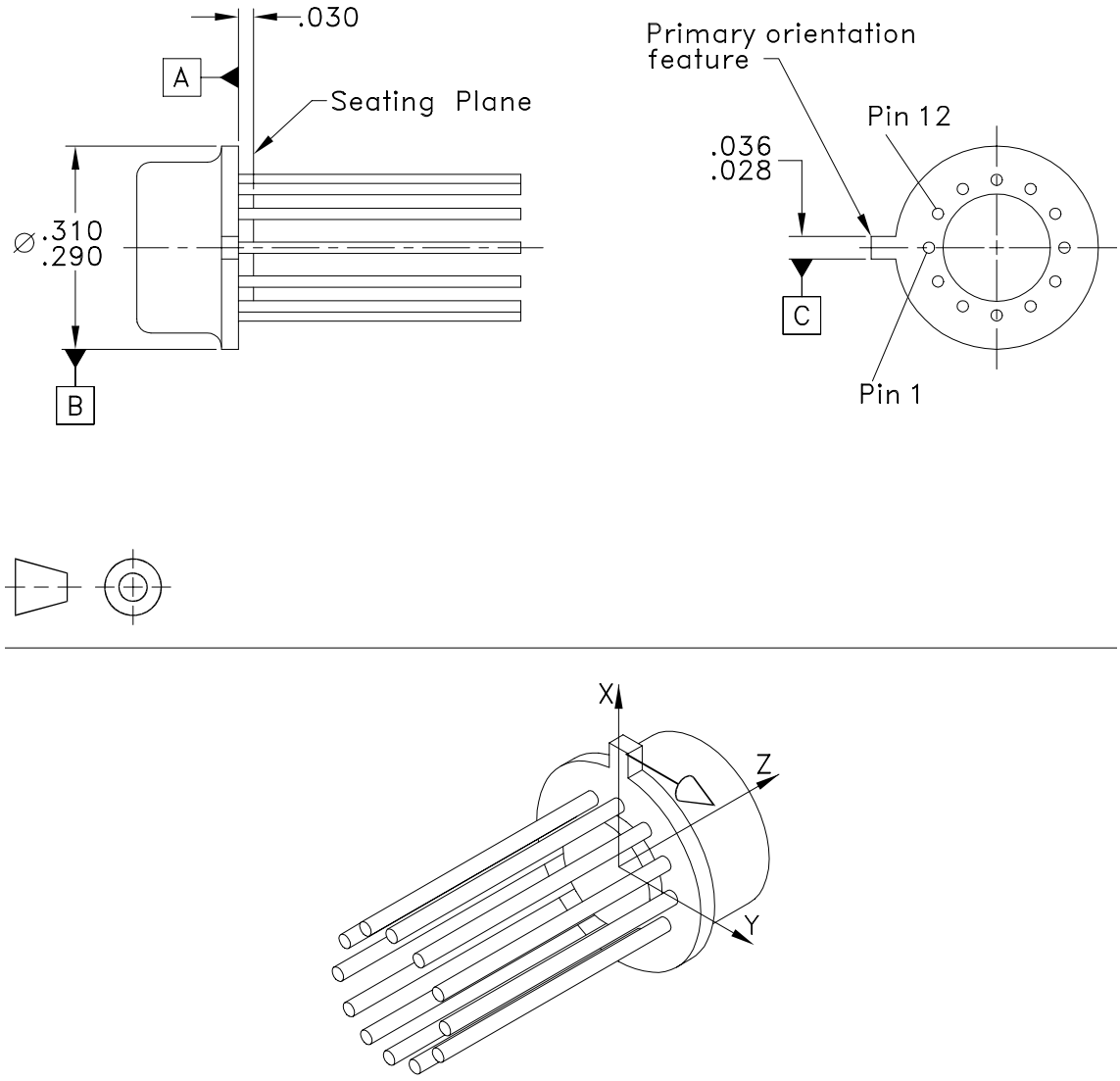


FIGURE 28 cylindrical through hole

FIGURE 29 Cyndrical through hole with orientation Symbology is an alternate representation of the part which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology.

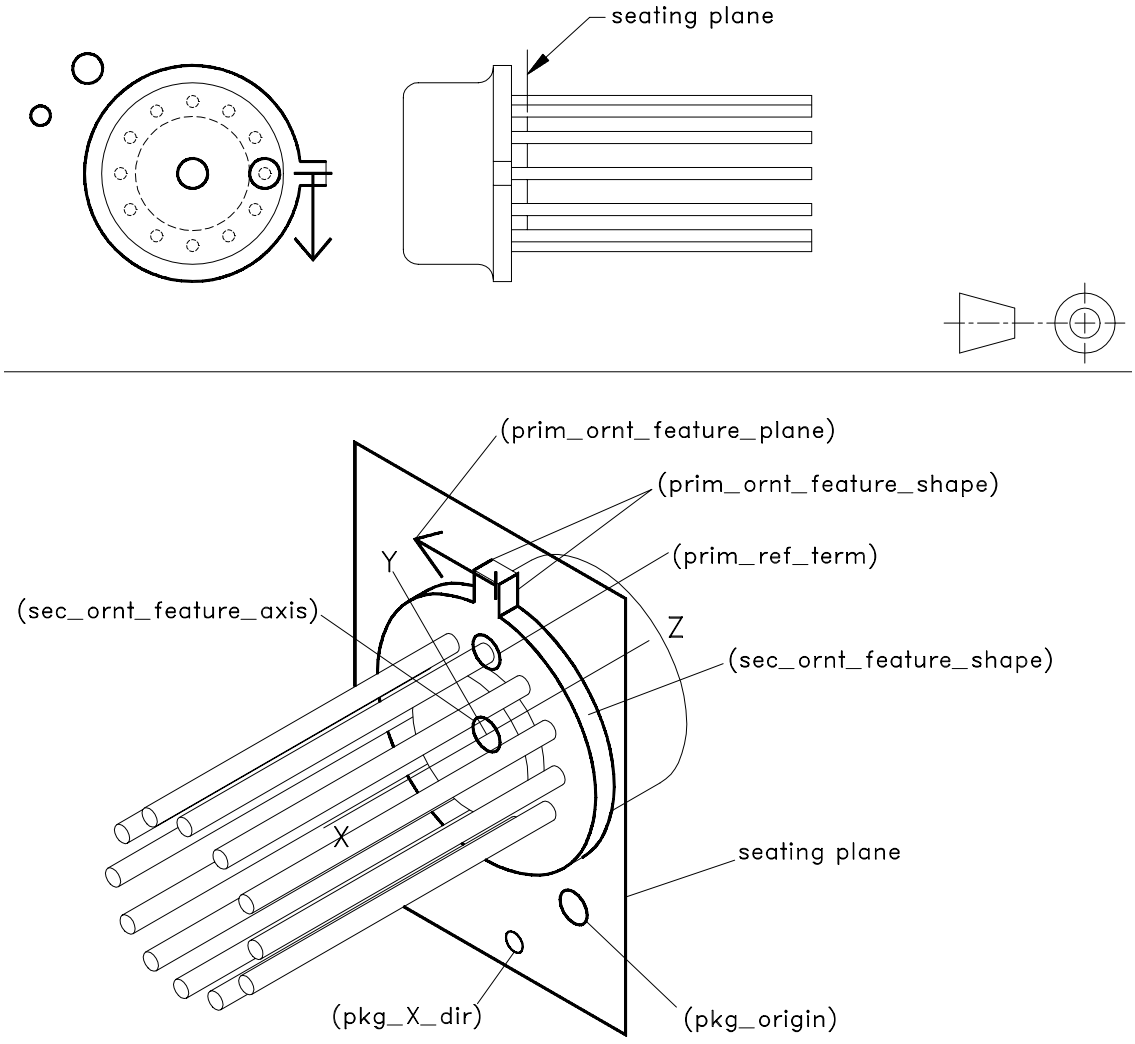


FIGURE 29 Cyndrical through hole with orientation Symbology[1]

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

2.6.8.6.7 Rectangular surface mount packages

The orientation features for this case are two opposing surface feature sets, a bottom surface feature and an identification feature. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

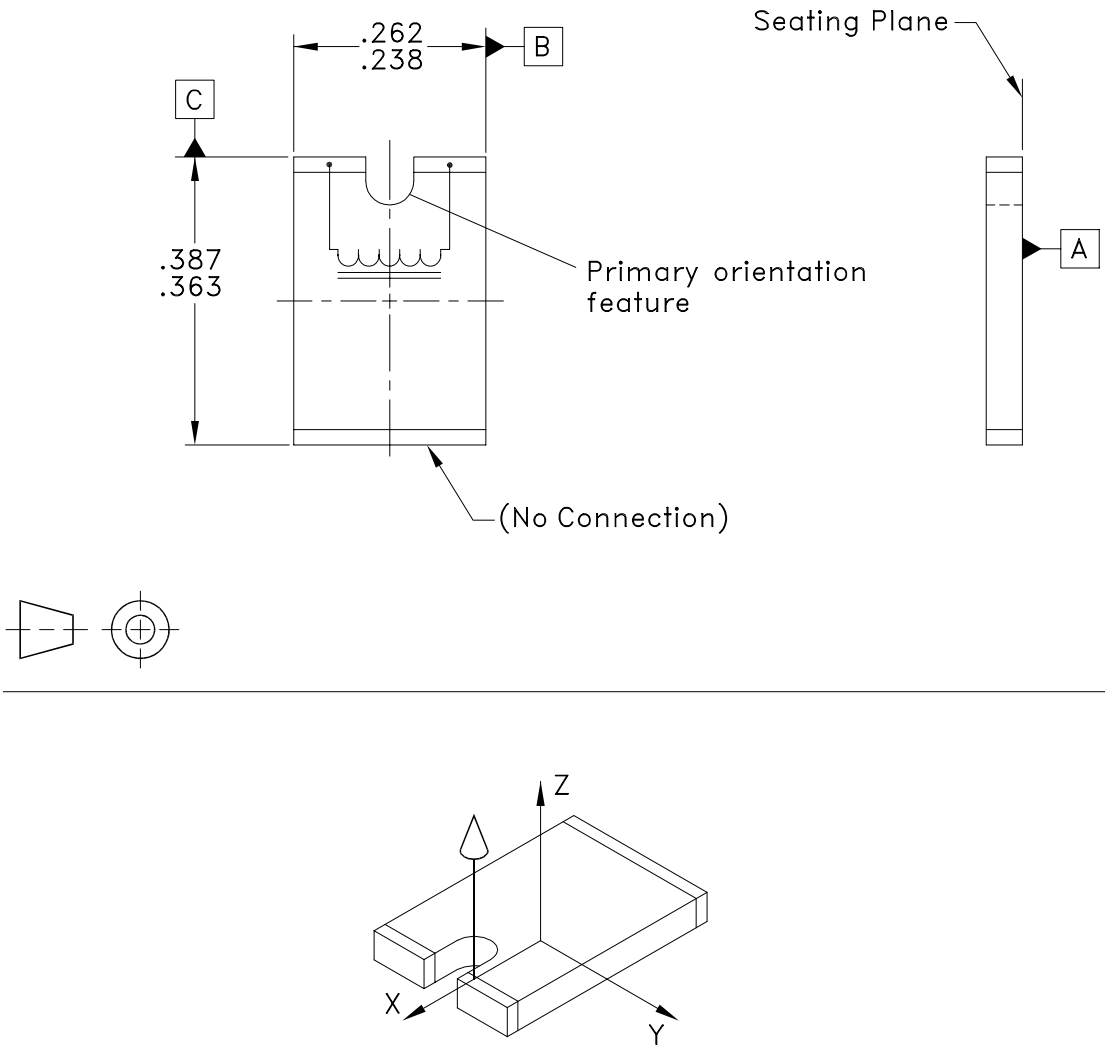
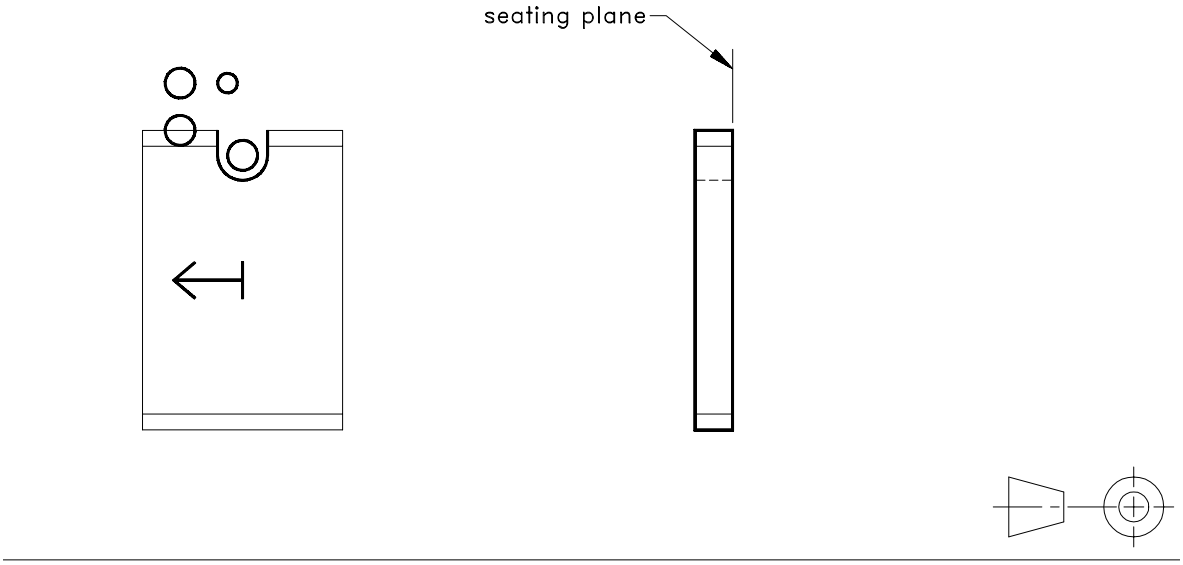


FIGURE 30 surface mount

FIGURE 31 Surface mount with orientation Symbology is an alternate representation of the part which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology.



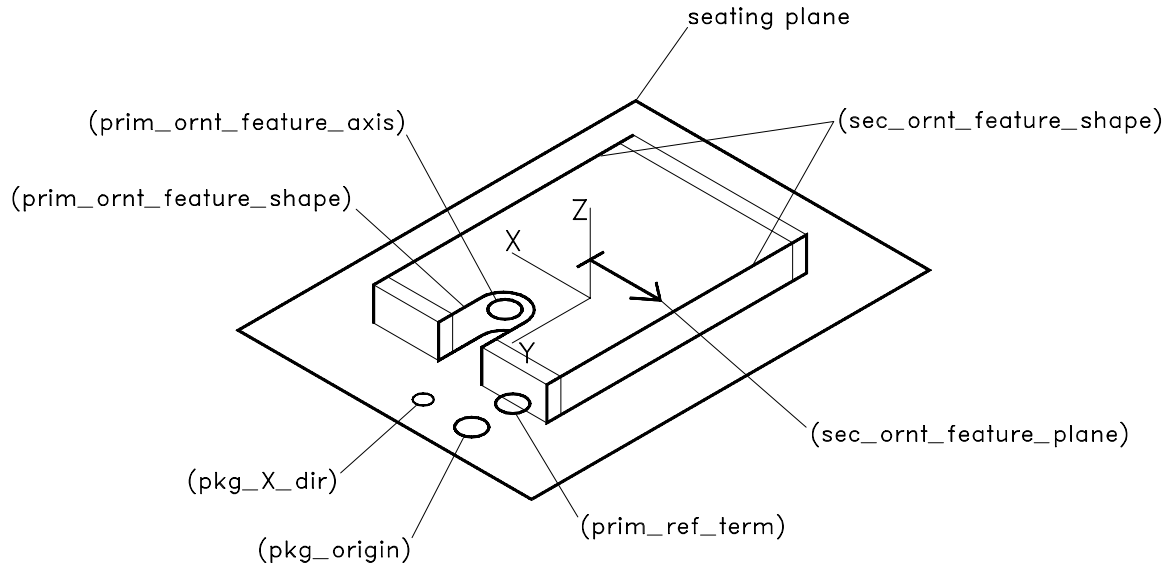


FIGURE 31 Surface mount with orientation Symbology[1]

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

2.6.8.6.8 Ball grid array packages

The orientation features for this case are two opposing surface feature sets, a bottom surface feature and an identification feature. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature. The Seating_plane should be parallel to the primary Datum_plane.

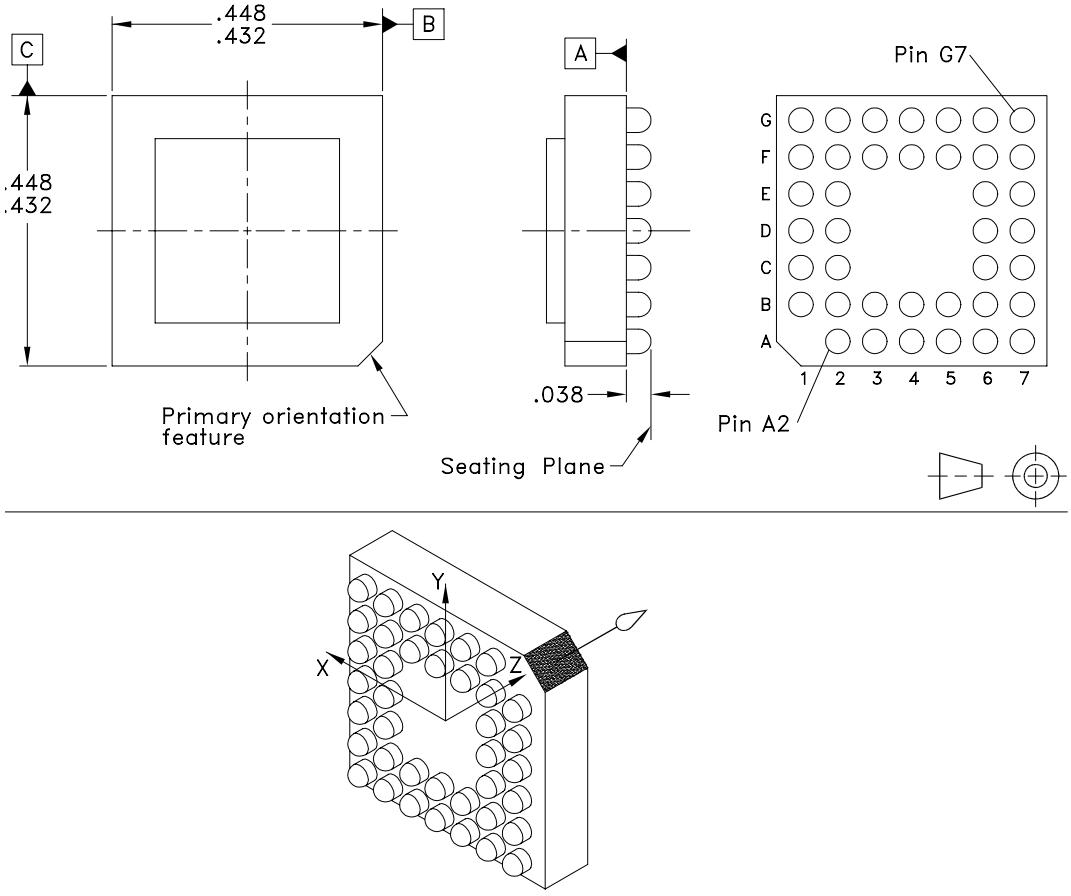


FIGURE 32 Ball Grid Array Illustration

FIGURE 33 Ball Grid Array with orientation Symbology is an alternate representation of the part which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology.

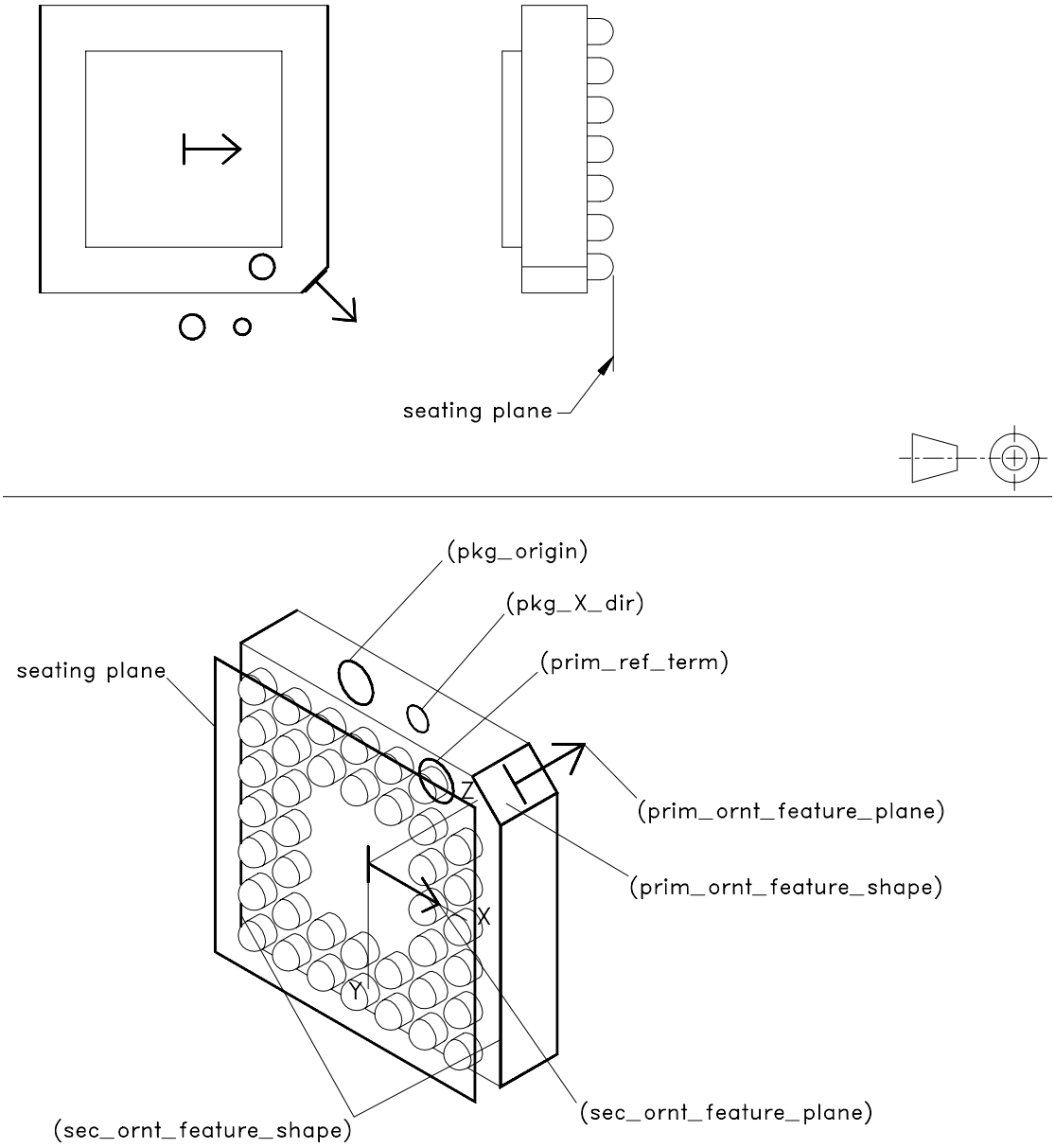


FIGURE 33 Ball Grid Array with orientation Symbology[1]

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

2.6.8.6.9 Rectangular top mounted surface mount leaded packages

The orientation features for this case are two opposing surface feature sets, a bottom surface feature and an identification feature. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

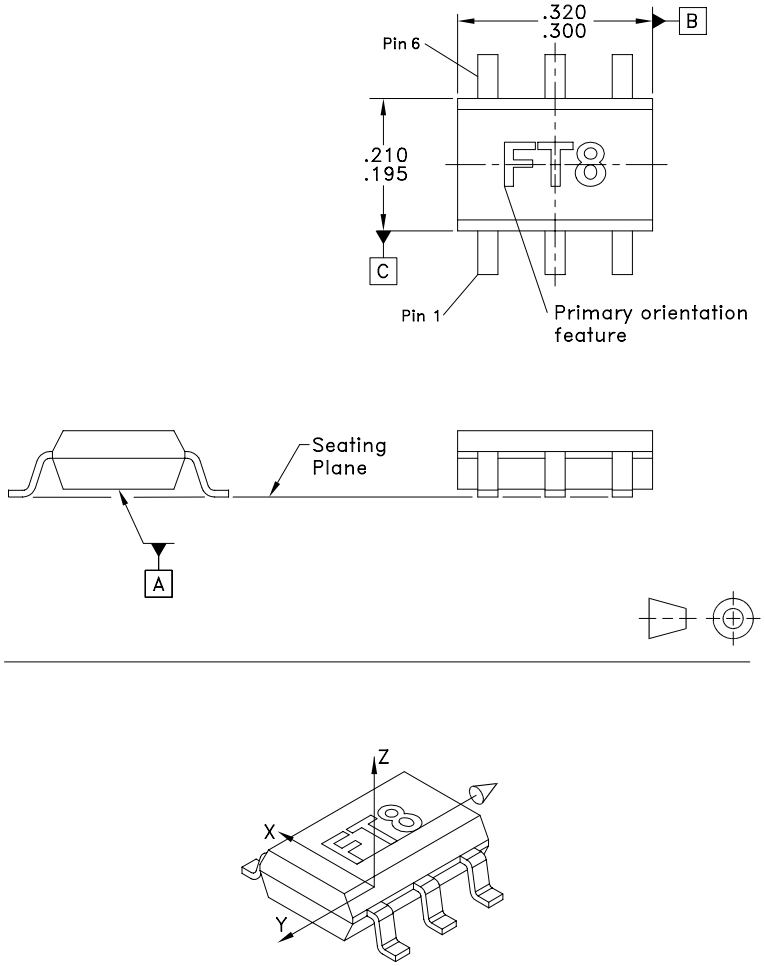


FIGURE 34 top mounted soic

FIGURE 35 Top mounted Soic with orientation Symbology is an alternate representation of the part which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology.

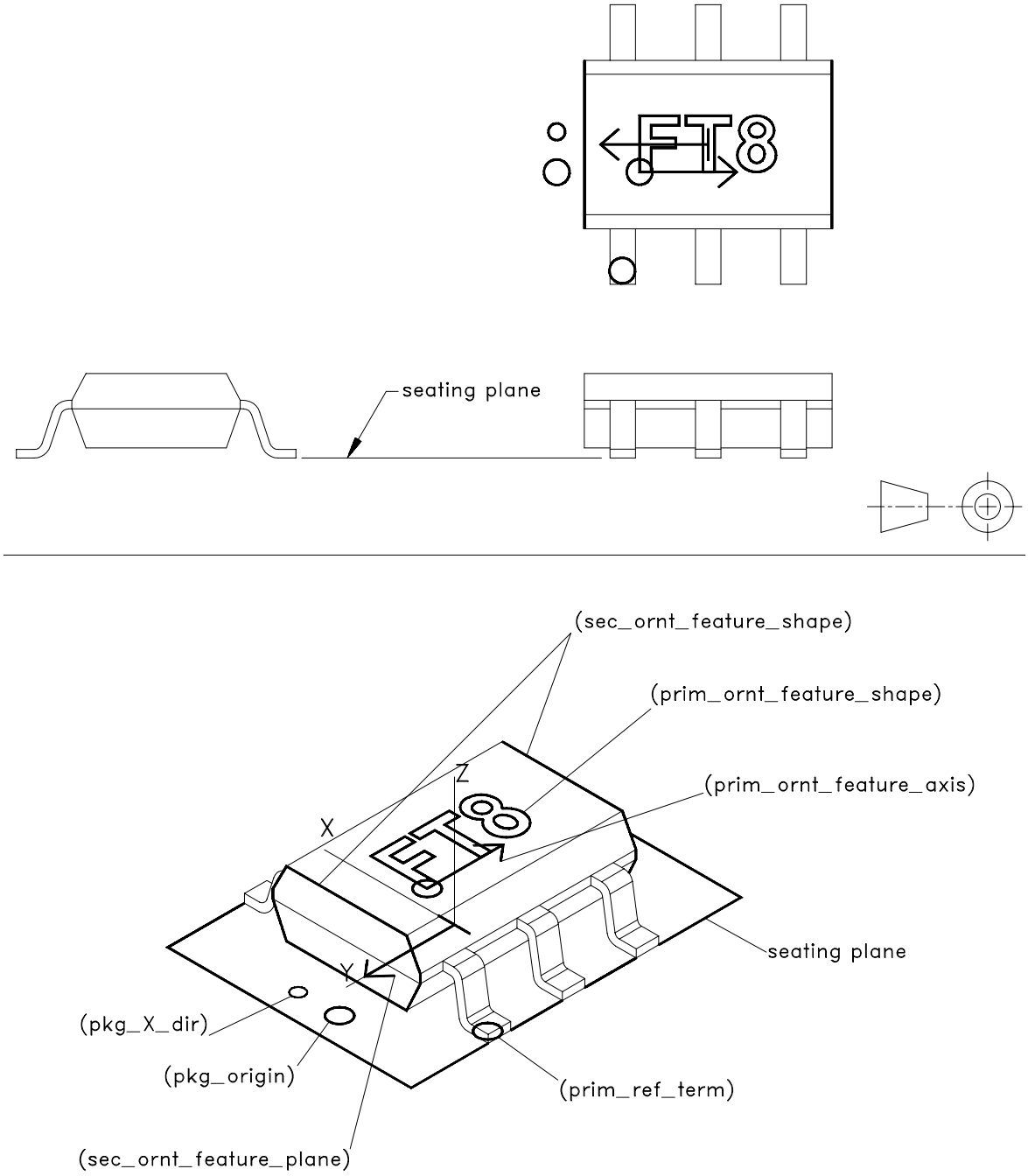


FIGURE 35 Top mounted Soic with orientation Symbology[1]

2.6.8.6.10 Rectangular cutout mounted surface mount packages

This surface mount packaged part is mounted in a cutout, with no changes in lead form, the part being oriented 180 degrees. The orientation features for this case are two opposing surface feature sets, a bottom surface feature and an identification feature. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

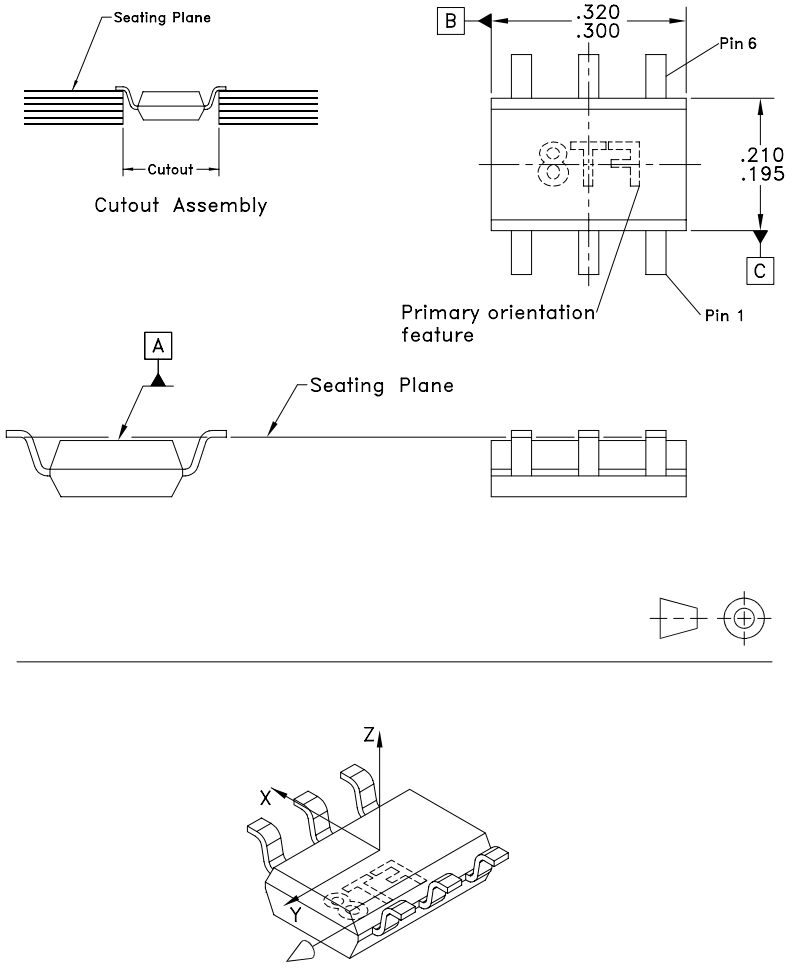
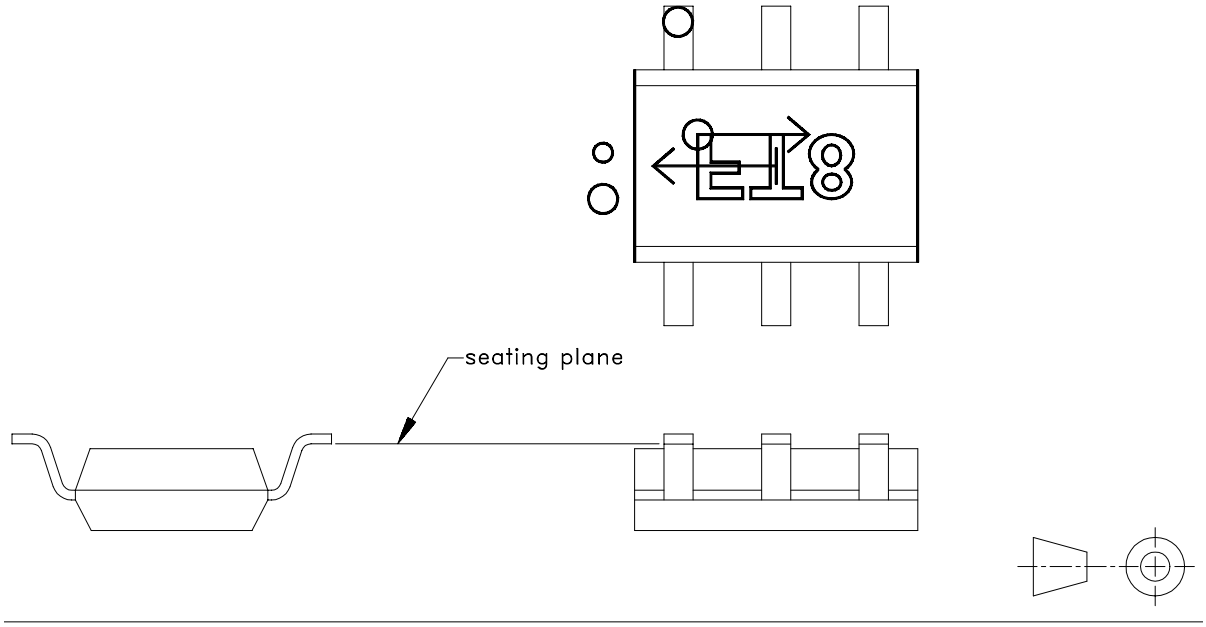


FIGURE 36 Cutout mounted Soic

[1] Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

FIGURE 37 Cutout mounted Soic with orientation Symbology is an alternate representation of the cutout mounted part which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology.



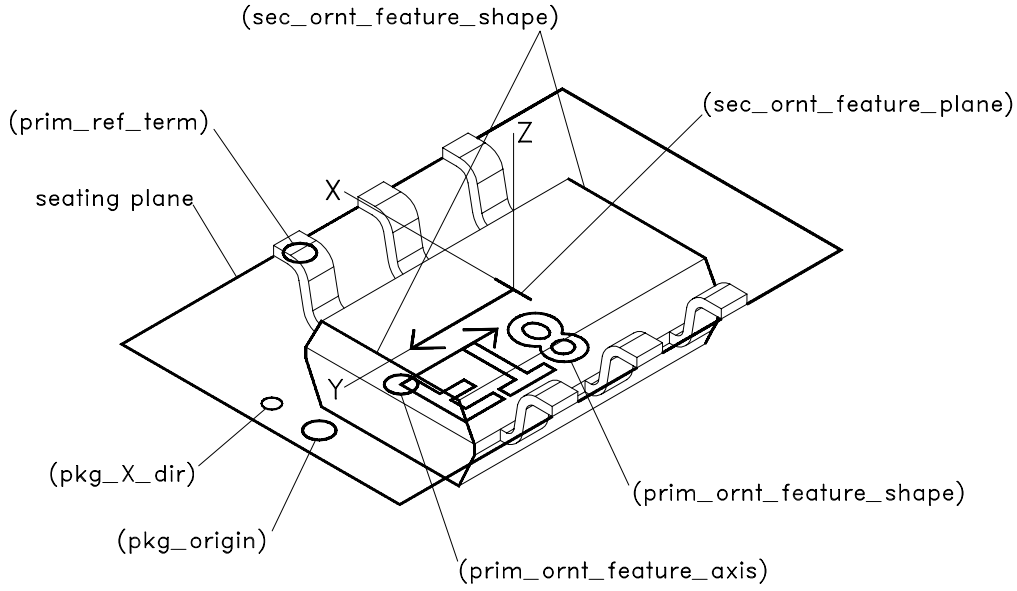


FIGURE 37 Cutout mounted Soic with orientation Symbology[1]

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

2.6.8.6.11 Right angle mount multi-pin connectors

The orientation features for this case are two opposing surface feature sets, a bottom surface feature and an identification feature. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

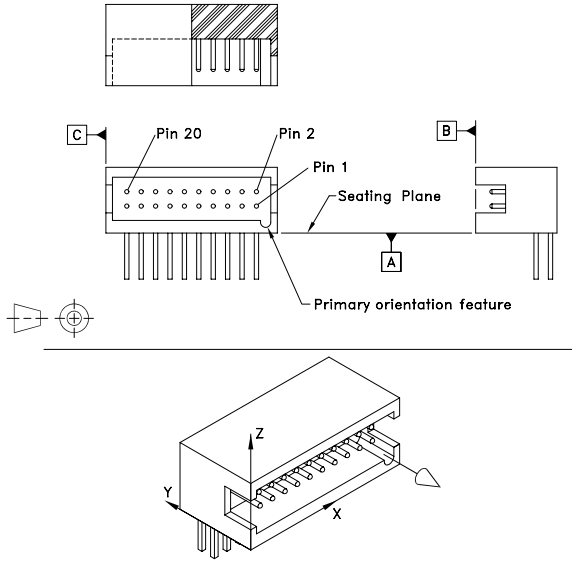


FIGURE 38 right angle 20 pin connector

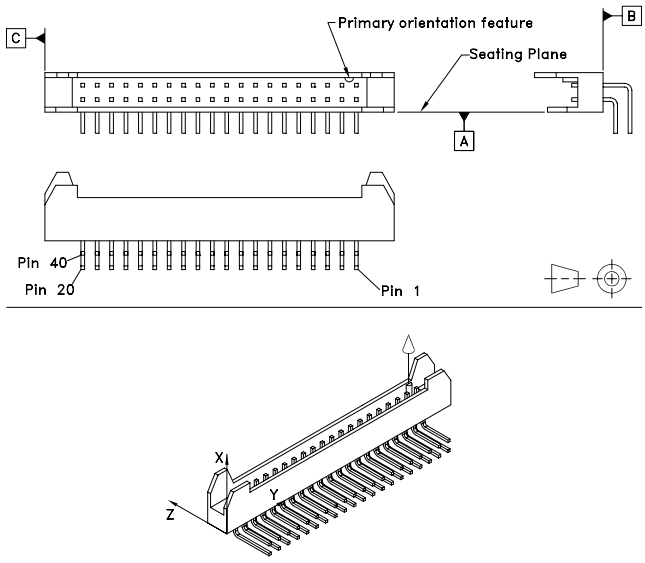
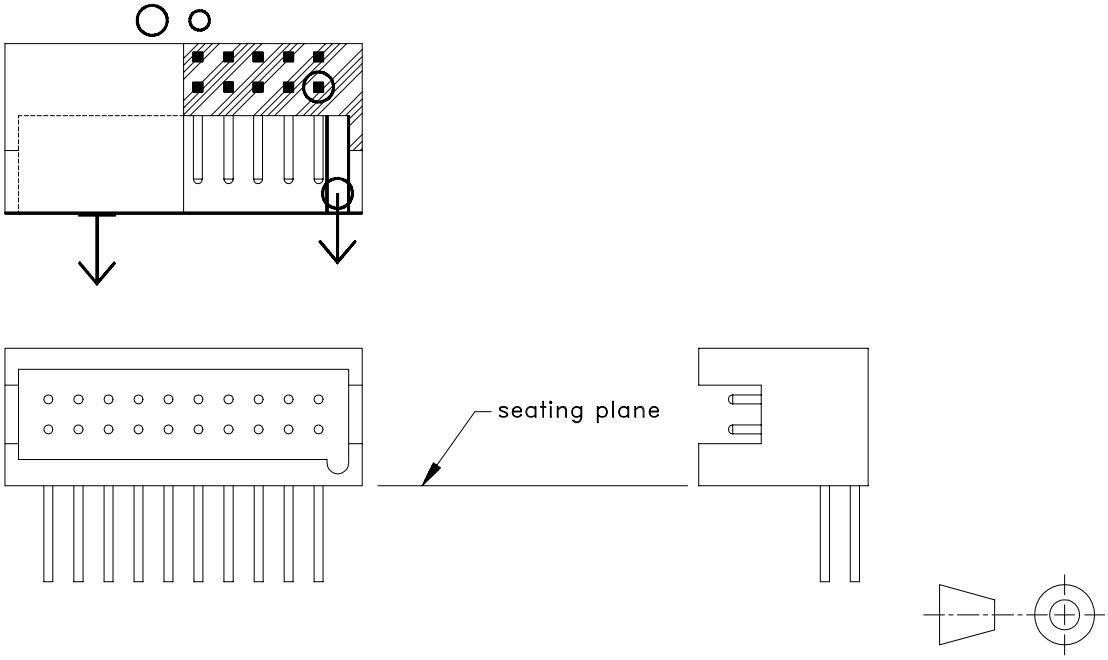


FIGURE 39 right angle 40 pin connector

FIGURE 40 Right angle twenty pin connector with orientation symbology and FIGURE 41 Right angle fourty pin connector with orientation symbology are alternate representations which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology



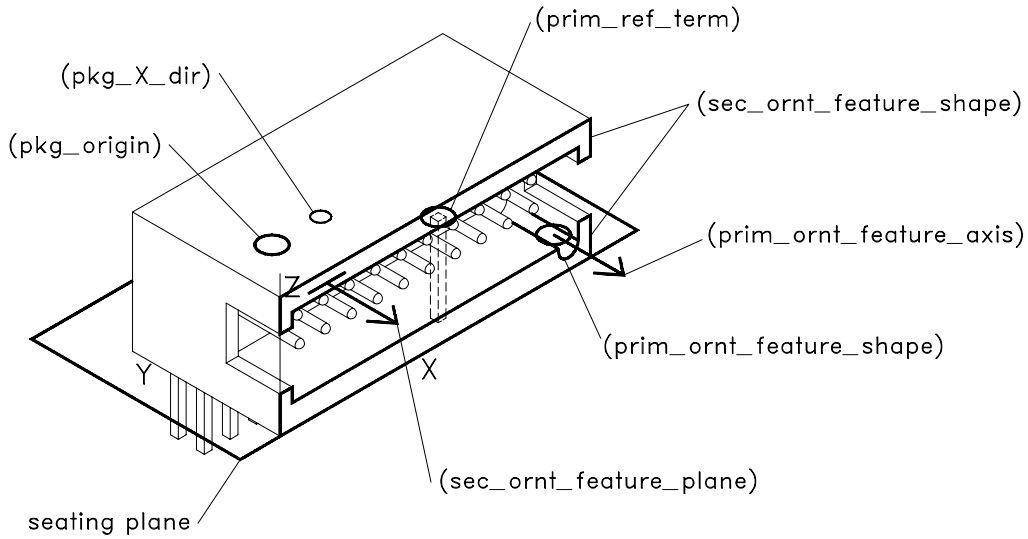
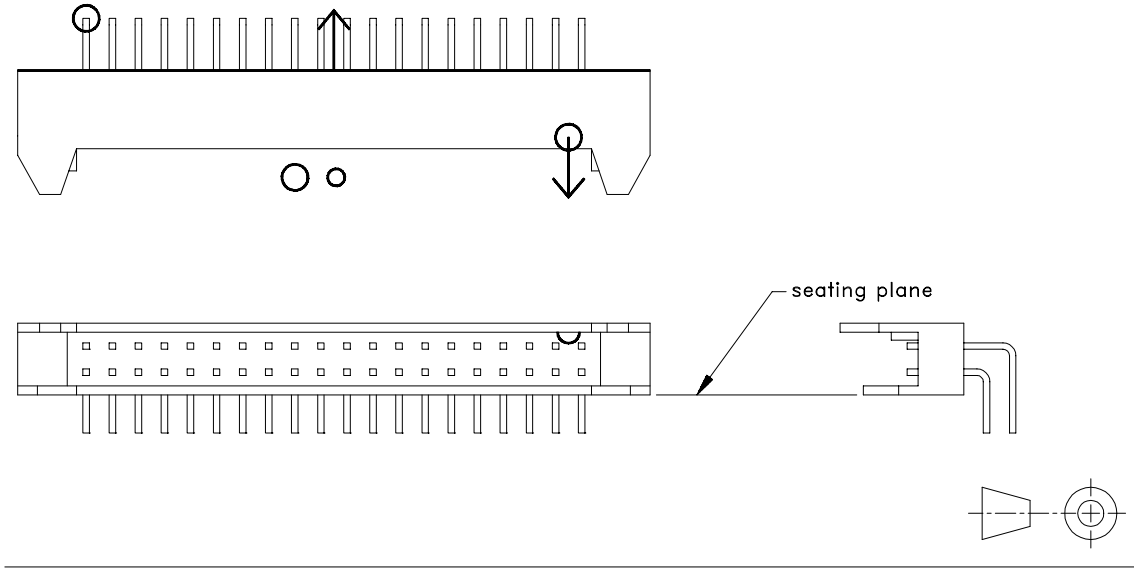


FIGURE 40 Right angle twenty pin connector with orientation symbology[1]

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity



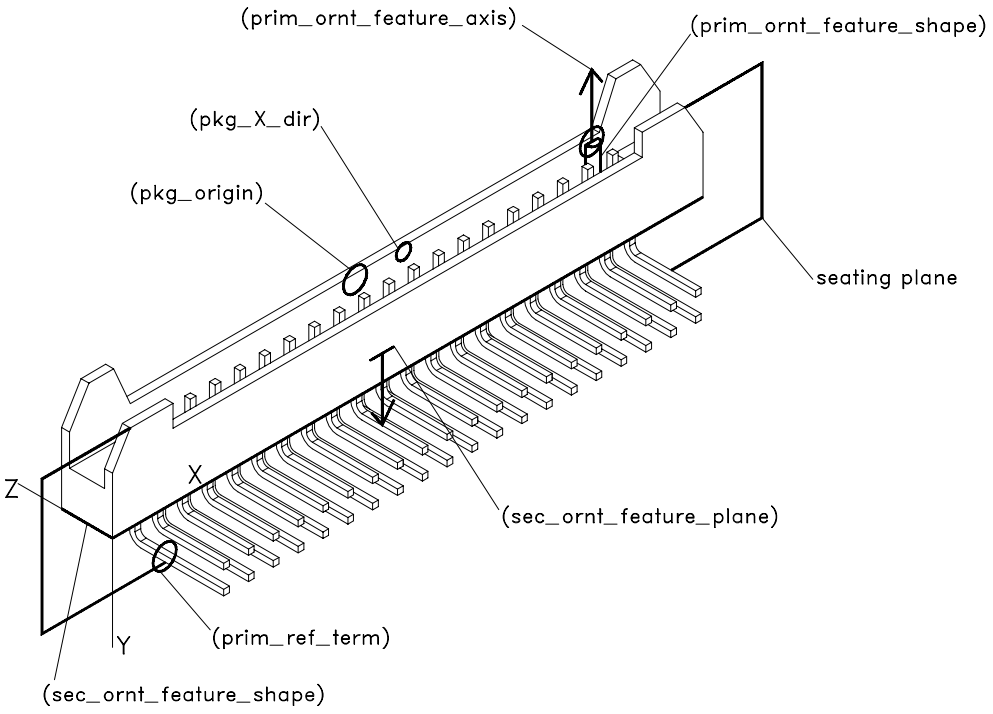


FIGURE 41 Right angle fourty pin connector with orientation symbology[1]

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

2.6.8.6.12 Right angle coaxial connectors

The orientation features for this case are two opposing surface feature sets, a bottom surface feature and an identification feature. The primary Datum_plane is associated with the bottom surface of the body, which is a Part_feature. The Seating_plane should be parallel to the primary Datum_plane. The Datum_axis associated with the Secondary_orientation_feature shall intersect the Primary Datum_plane at right angles.

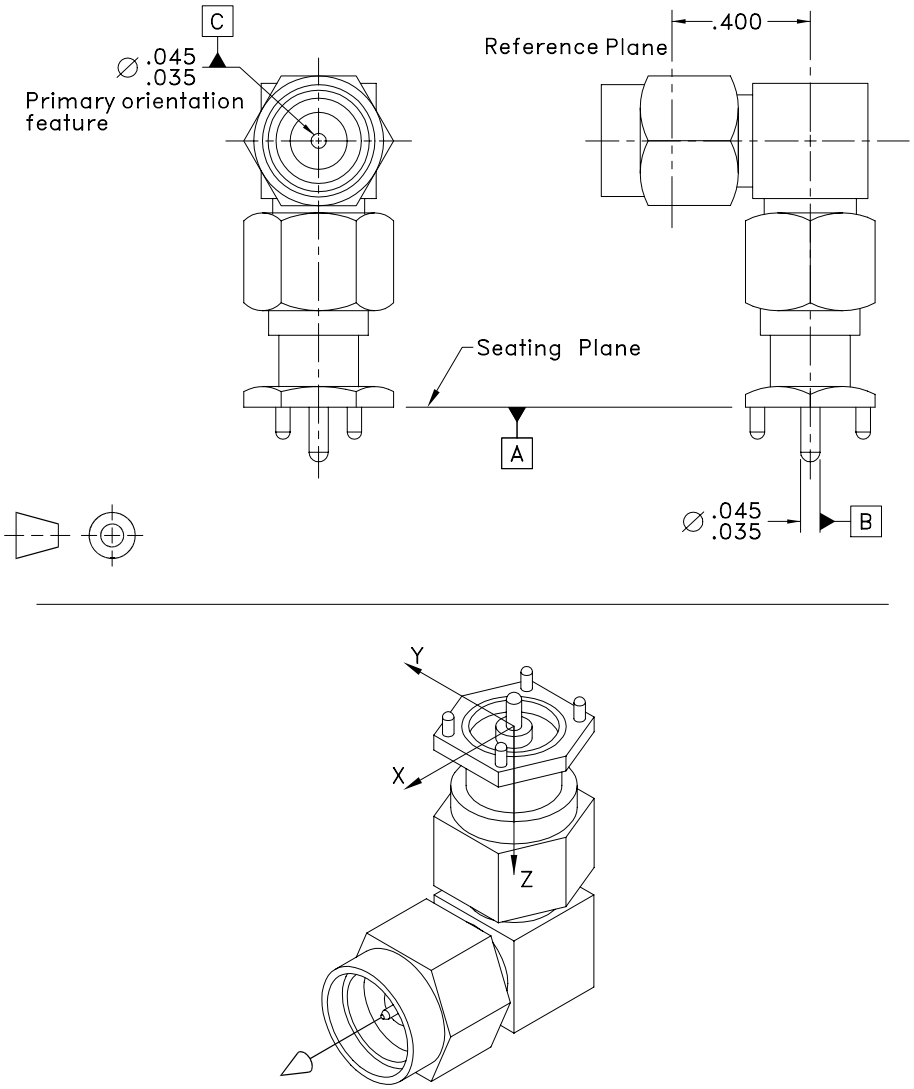


FIGURE 42 Sma Connector

FIGURE 43 Sma Connector with Layer Symbology is an alternate representation of the Sma connector which includes graphic symbols in the 3D model that are in accordance with the layer based representation symbology. [1]

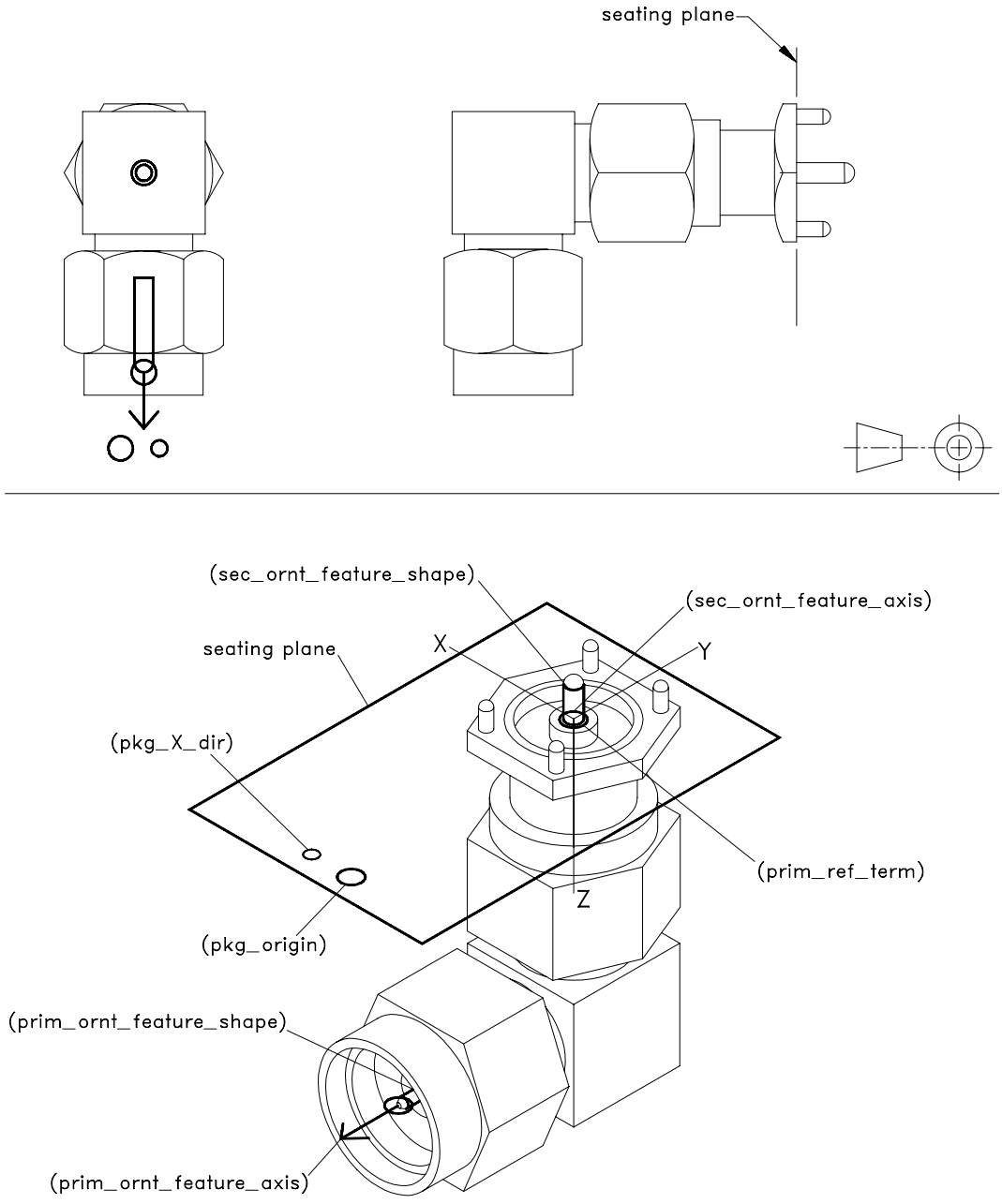


FIGURE 43 Sma Connector with Layer Symbology

1. Package origin and X dir symbology are offset from actual origin for clarity. Non standard abbreviations are used for simplicity

2.6.8.6.13 Non Orientable Connectors

In some cases, the orientation information for a connector is supplied by its location and orientation on the substrate, or more accurately, by the location and orientation of the Component_termination_passages or by the location and orientation of the Stratum_feature_template_components on the substrate that are there to support that connector. In this case, the orientation information cannot be populated sufficiently in Package. In AP 210, the orientation information supporting Package Application object is optional to support this case.

2.6.8.6.14 Symmetrical Components

In some cases (e.g., two terminal leadless components) it will be necessary to make arbitrary decisions in the assignment of orientation features and associated datum. The recommendation is to always ensure the principal axis is a datum and that GD&T rules for creation of datum reference frame be followed. The defining data is not found in the component specification data from the manufacturer, so in this case, the library data is the defining data source.

2.6.8.6.15 Composite Illustrations

The following figures are the four composite illustrations cited earlier.

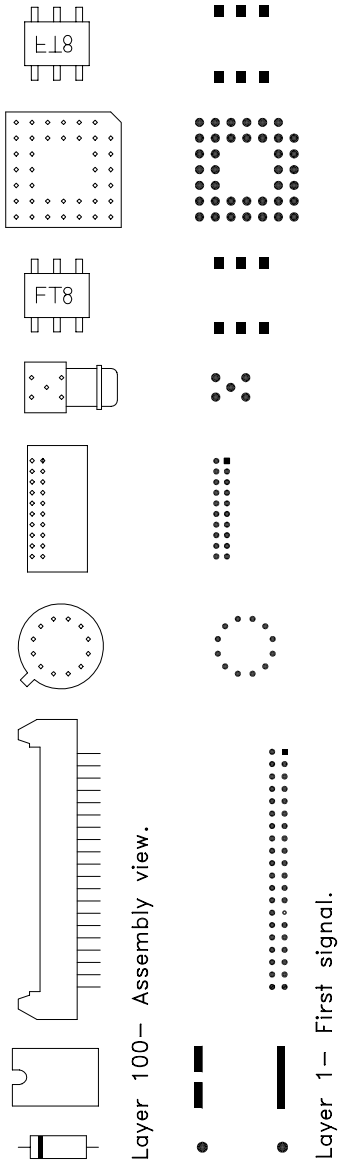


FIGURE 44 Assembly and Land shapes

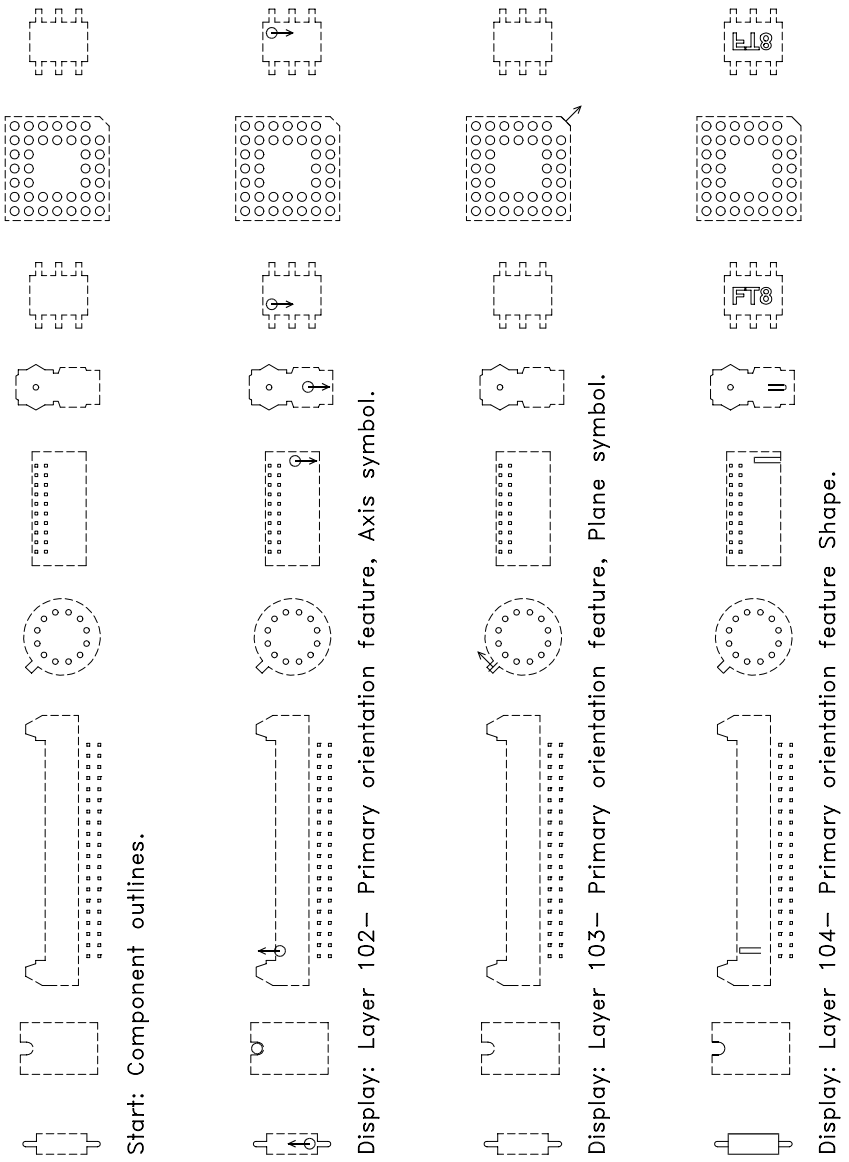


FIGURE 45 Component outline and Primary orientation feature data

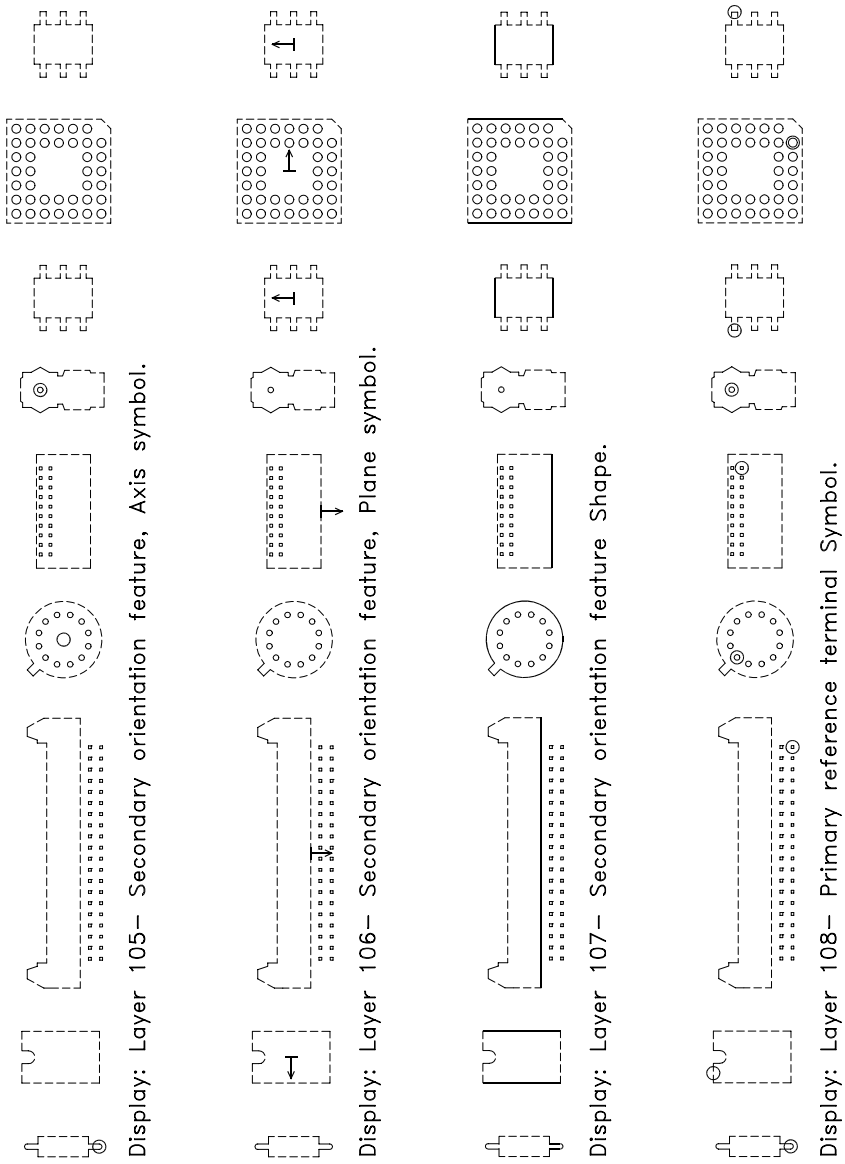


FIGURE 46 Secondary orientation feature and Primary_reference_terminal data

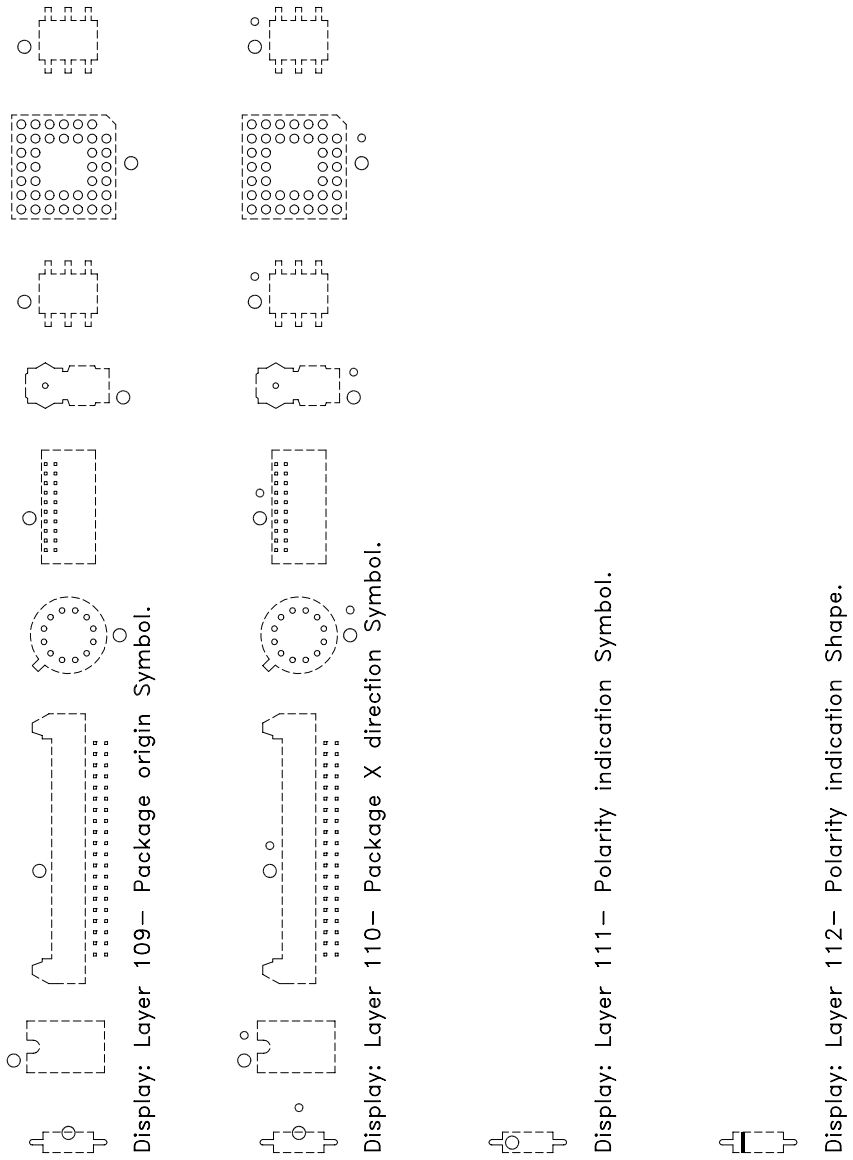


FIGURE 47 Package origin and Polarity_indication_feature data

2.6.8.7 package alternates used by an enterprise

In this context, a package alternate means an alternative graphic symbol that represents certain useful geometric properties of a part as defined in a CAD system library, which may be specific to an enterprise, or which may be industry accepted practice.

2.6.8.7.1 Package pick shape

The case considered is that there is a layer set aside in the 2D CAD system for a shape that represents the shape of the package body minus tolerances. This shape is a smaller area than the actual package, and is intended to provide a margin of error for pick and place machines.

2.6.8.7.2 package shape modification alternates

Enterprise modifications to the shape of parts received from a supplier are sometimes represented merely by package alternates (of the same part) in a design CAD system. Sometimes they are tracked as different products (supported by ARM Application objects Ee_product, Ee_product_version, Altered_packaged_part, Altered_package and Altered_package_terminal) by CAD, Manufacturing Resource Planning (MRP) and CIM systems. There appears to be no standardization of treatment in the enterprise application of CAD systems to part alterations.

Enterprise modification may include a different form or fit (e.g., an axial part received with straight leads has leads bent into a hairpin style for insertion. The resulting part is a different part because it no longer has the same form or shape as the original part). Each Altered_package has a complete set of Package properties, but identifies the base Package as well. Each Altered_package has its own seating plane, so "tulip" ("tulip" refers to mounting a can upside down in a passage in the substrate and attaching the leads to lands on the surface rather than using the usual "through hole" lead insertion.) mounted JEDEC TO-99 cans defined by an instance of Altered_package would not use the same geometric model or seating plane as the through hole variant.

The enterprise using the CAD system considered herein provides an alternate shape for a package in the enterprise instance of the native CAD library specifically for the purpose of representing a component mounted in the reversed position on the top of the substrate. As an example, consider an enterprise library with a JEDEC TO-5 package. The library information includes the location of all terminals, their names, and the fact that one of the terminals is the ARM Application object Primary_reference_terminal. The library information also includes the information that the tab is an ARM application object Primary_orientation_feature, and its location. The enterprise has decided that TO-5_alt1 is an accurate indications of the mounting information for through-hole insertion where the leads may be slightly spread. For "tulip" mounting style, the terminals are to be displayed with an alternate shape graphically indicating the lead arrangement desired and the ARM Application objects Altered_package and Altered_packaged_part are used. More importantly the seating plane is oriented differently with respect to the package, and the terminals are re-arranged, leading to a change in at least Fit properties. The package mounting is labeled ’TO-5-tulip1’. The enterprise assigns a different part number (using Ee_product, Ee_product_version and Altered_packaged_part) to the product associated with TO-5-tulip1 indicating it is not the same part as TO-5_alt1, but tracks the dependent relationship with the ARM Application object attribute Altered_packaged_part.base_packaged_part.

Pre-processor recommendation: The enterprise needs to define policies to extract package alternate data associated with altered parts and convert it into the AIM equivalents of the ARM Application objects Ee_product, Ee_product_version, Altered_package, Altered_packaged_part and Altered_package_terminal.

Post-processor recommendation: The inverse of the pre-processor recommendation applies.

2.6.8.7.3 mounting side design presentation alternates

If the CAD system considered herein did not support mirroring, then package alternates would be provided so that the features of components mounted on the bottom of the substrate are correctly presented to the designer when looking "down" through the board. In the CAD system case considered herein there is no alternate required.

The following table identifies possible combinations of package case styles used, the mounting side, and shows the value of the transformation matrix for each mounting side.

Package case style selection table

Substrate side component is mounted on	Component lead attachment method	Package case style	Mirror shape value	AP 210 Transformation matrix value (T)
primary	through hole insertion	TO-5	false	1.0
primary	surface mount tulip	TO-5-tulip	false	1.0
secondary	through hole insertion	TO-5	true	-1.0
secondary	surface mount tulip	TO-5-tulip	true	-1.0

This table is a combination of data from the library and from the component instance in the assembly. The key to this table is {Substrate side}.

Pre and Post processor responsibilities

Enterprise library recommendation:

It is recommended that enterprises that require mirroring information in the package alternate in the the enterprise library provide a package property with a property name of "ISO_10303-210_mirror_shape" to support the "mounting side design presentation alternates". The recommended values for the property are "true", "false". The value is populated according to the table in the section above. Note that the existence of this property is dependent on whether enterprise policies require a separate package alternate for each substrate side when mounting components on both sides of the substrate. The reason this property is needed is not to indicate whether or not the component is mounted on the bottom. It is so that a processor can query the library to determine that the shape is to be mirrored. It is expected that in the majority of cases, this enterprise specific property will be unnecessary since most enterprises will choose to let the pre-processor extract the mirroring information directly from the CAD design definition.

Pre-processor:

In the 2D CAD case considered herein, the pre-processor is responsible for converting from the CAD and enterprise specific library definition to the standard definition in AP 210. The mounting_surface shall always be specified. The shape of the unmirrored "mounting side design presentation alternate" should be captured in the "nominal material condition", "design", "manufacturing" state of the ARM Application object Physical_unit_planar_shape.

Post-processor:

Populate the unmirrored "mounting side design presentation alternate" based on the "nominal material condition", "design", "manufacturing" state of the ARM Application object Physical_unit_planar_shape.

2.6.8.7.4 assembly drawing alternates

Package alternates may be provided for the purpose of creating drawings only (i.e., a view of an assembly will require an image showing locating features on a component visible from the operator or vision system when the component is installed). The creation of drawings is often treated as a post-process which includes changing the package alternate from the design usage to the drawing generation usage. The symbolic information must be coordinated with the location properties.

AP 210 provides support to unambiguously associate an oriented graphic image symbol (**shape_representation**) with the component location so that an assembly drawing can be generated directly from an AP 210 data repository. The ARM Application objects Primary_orientation_feature, Polarity_indication_feature, and Primary_reference_terminal may be included in the assembly drawing alternate graphic. This graphic includes in it geometry representing features of the component intended to represent its appearance in an assembled condition. The appearance of this graphic is very enterprise specific although the information in the graphic can be determined by querying the Application object Usage_concept_usage_relationship. The shape and detailed semantic content of this graphic is not defined by the standard. Composite package shape described earlier may be used for this application.

Pre-processor recommendation: The enterprise needs to define policies to extract package alternate data associated with assembly drawing alternates and convert it into the ARM Application object Physical_unit_planar_shape with a purpose of "assembly view" and form exchange agreements to formalize the usage.

Post-processor recommendation: The inverse of the pre-processor recommendation applies.

2.6.8.8 Additional Information for Modeling Substrate Mounted Connectors

AP 210 provides explicit modeling of the fact that a substrate mounted connector is designed with an interface that mates with a substrate and a side that interfaces with an external system. In AP 210 the explicit model is accomplished by logically splitting pin of the connector into two terminals: Packaged_part_join_terminal and Packaged_part_interface_terminal. The two terminals are specified as being for the same pin (i.e., a short) by an instance of Packaged_Connector_Terminal_Relationship. Most 2D CAD systems will ignore the Packaged_part_interface_terminal as it does not effect the conductive pattern on the printed wiring board. Orientation information capture is described previously.

2.6.9 Conversion Between 3D CAD system and AP 210 3D model

Shape_representation should include **axis2_placement_3d** as origin. The preferred transformation in AP 210 is to use **mapped_item** and **representation_map**. **Shape_aspect** identifiers should be persistent. Other items may be derived from previous sections of this document. If the native CAD system does not support shape_aspect it is recommended to use layers capability in order to get the data out of the CAD system. Once the layer data is available in AP 203, it can be translated into shape_aspect subtype in AP 203 with the attribute data applied.

2.6.9.1 Non-planar layout

AP 210 provides for direct support of non-planar layout through the 3D capabilities.

2.6.9.2 Creating Source Package model data

A method for obtaining shape data for Packages in AP 210 is to model the Package_body and Package_terminals as individual piece parts and model the Package as an assembly. The 3D CAD AP 203 output translator would then be executed. The resulting AP 203 file is then converted to an AP 210 file using the knowledge that the context is really a single Package and the piece parts are the features (Package_body is a type of Part_feature). In the3D CAD context, the feature geometry can be defined once, given a part number and then referenced as needed. The features that need pin numbers or type assignments (e.g, Primary_orientation_feature) can be identified with the next_assembly_usage_occurrence reference_designator and id attributes. The 3D CAD may not use mapped_item and representation_map to locate the piece parts in an assembly but may use another option from AP 203. An enterprise procedure is developed for the particular 3D CAD system and AP 203 translator combination, and appropriate conversion s/w written. Since the AP 203 model is based on an EXPRESS schema and AP 210 is also based on an EXPRESS schema, it is recommended to use EXPRESS-X to code the conversion algorithm.

2.6.9.3 Creating Source Connector model data

Reference the section describing conversion between 2D CAD and AP 210 for connectors for the introductory information. The Packaged_part_interface_terminal or other feature on the interface may be used in the system for orientation purposes. If so, then that feature is designated as a Primary_orientation_feature. A key feature of this capability is to provide enough information so that network listing of actual connections may be extracted from 3D CAD assembly context.

Enterprise procedures and software used to create Packages is directly usable to support Connectors, but will need to be extended to add the knowledge of Packaged_part_interface_terminal and Packaged_Connector_Terminal_Relationship.

2.7 Component_placement_restriction_assignment

This section references the updates in the mapping table accomplished since the DIS document. Reference the graphic in ***component_placement_restriction_assignment.pdf***

2.8 Conductive_interconnect_element

This section references the updates in the mapping table accomplished since the DIS document. Reference the graphic in ***conductive_interconnect_element_expl.pdf***

2.9 Interconnect_module_constraint_region

This section references the updates in the mapping table accomplished since the DIS document. Reference the graphic in ***Interconnect_module_constraint.pdf***

Appendix A: Sections to be added

- Padstacks
- Land Patterns
- Schematics
- Data Element Type (DET)
- Gerber Replacement
- IPC 350 Compatibility
- Intermediate Data Format (IDF) compatibility
- GDSII Compatibility
- Functional netlist
- Physical netlist (elaboration)
- Pin Mapping
- Swapping
- External Definition
- Test fixture
- Electronic Data Sheet
- Electrical/Mechanical Integration
- PCB Panelization
- Assembly Panelization
- Design Reuse
- Board Stackup Definition
- Default PCB Tolerance Definition
- Layout Router Interface
- RF Layout
- Printed Transformer
- Printed Connector
- Fabrication Technology
- Assembly Technology
- Design Validation
- Wiring and Cabling